BERLINER MAUER 1961 - 1989
Windstraße
THE TEMPLE BAR
Bis zu 5% Bonus für treue Kunden
SICHERN SIE SICH IHREN TREUEBONUS.
2b
SAARBRÜCKEN
EINSTEIN
MAXWELL LN

2007

Camera-Based Document Analysis and Recognition

Proceedings of the

Second International Workshop on Camera-Based Document Analysis and Recognition

September 22, 2007

Grand Hotel Rayon, Curitiba, Brazil

Edited by

Koichi Kise
Osaka Prefecture University, Japan

David S. Doermann
University of Maryland, USA

CBDAR 2007

CBDAR2007 Sponsors:

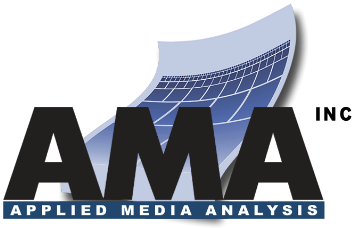


Table of Contents

Oral Papers

Section I. Text detection

Section II. Retrieval

Section III. Dewarping

Poster Papers

Demo Papers

* represents a related paper

Contest

* represents a related paper

Preface

The pervasive use of camera phones and hand-held digital cameras has led the community to identify document analysis and recognition of digital camera images as a promising and growing sub-field. Constraints imposed by the memory, processing speed and image quality of these devices are leading to new interesting problems to be addressed.

Following the success of the first CBDAR2005 workshop in Seoul, Korea, 2007 Workshop on Camera Based Document Analysis and Recognition (CBDAR) is being held in conjunction with the International Conference on Document Analysis and Recognition (ICDAR) in Curitiba, Brazil. The goal is to bring together researchers to present cutting edge applications and techniques and contribute to discussions on future research directions.

This proceedings contains a written archive of the papers to be presented. This year, we had 21 full paper submissions, from which 7 were accepted for oral presentation. They fall into three general areas of research – text detection, document retrieval and dewarping. In addition, 12 posters will be presented on various algorithms and applications, and 11 organizations will participate in the demonstration session. Each presenter was invited to present a full paper describing their contribution, and they are contained in the proceedings.

This year, Prof. Thomas Breuel and Mr. Faisal Shafait from the University of Kaiserslautern are organizing a page dewarping contest. One of the major research challenges in camera-captured document analysis is to deal with the page curl and perspective distortions. Current OCR systems do not expect these types of artifacts, and have poor performance when applied directly to camera-captured documents. The goal of page dewarping is to flatten a camera captured document such that it becomes readable by current OCR systems. Page dewarping has triggered a lot of interest in the scientific community over the last few years and many approaches have been proposed, yet, until now, there has been no comparative evaluation of different dewarping techniques. Results from the contest will be presented during the workshop.

Finally, we would like to sincerely thank those who are helping to ensure this workshop is a success. The ICDAR organizing committee – Prof. Robert Sabourin (Co-Chair), Prof. Kazuhiko Yamamoto (Workshop Chair), and Dr. Luiz Oliveira (Conference Manager) have been extremely helpful in making sure the workshop venue and scheduling were prepared. Prof. Masakazu Iwamura for the logo design, maintenance of web pages and mailing lists, and the preparation of the proceedings. The CBDAR program committee for reviewing and commenting on all of the submission we received. And the financial sponsors of the workshop: Applied Media Analysis, APOLLO, Hitachi, NEC, Osaka Prefecture University, Ricoh, and the University of Maryland. We thank you all.

We sincerely hope that you enjoy the workshop and that these proceedings provide an archival snapshot of this cutting edge research.

CBDAR2007 Co-Chairs
Koichi Kise and David Doermann

CBDAR2007 Program Committee

Co-Chairs

Koichi Kise, *Osaka Prefecture University, Japan*

David Doermann, *University of Maryland, USA*

Program Committee Members

Thomas Breuel, *DFKI, Germany*

Andreas Dengel, *DFKI, Germany*

C.V. Jawahar, *IIIT, India*

Jonathan Hull, *Ricoh Innovations, USA*

Soo Hyung Kim, *Chonnam National University, Korea*

Majid Mirmehdi, *University of Bristol, UK*

Gregory Myers, *SRI International, USA*

Shinichiro Omachi, *Tohoku University, Japan*

Shuji Senda, *NEC, Japan*

Jun Sun, *Fujitsu R&D Center, China*

Chew Lim Tan, *NUS, Singapore*

Seiichi Uchida, *Kyushu University, Japan*

Christian Wolf, *INSA de Lyon, France*

Assistant

Masakazu Iwamura, *Osaka Prefecture University, Japan*

Section I

Text detection

Font and Background Color Independent Text Binarization

T Kasar, J Kumar and A G Ramakrishnan
Medical Intelligence and Language Engineering Laboratory
Department of Electrical Engineering, Indian Institute of Science
Bangalore, INDIA - 560 012
tkasar@ee.iisc.ernet.in, jayantkmishra@gmail.com, ramikag@ee.iisc.ernet.in

Abstract

We propose a novel method for binarization of color documents whereby the foreground text is output as black and the background as white regardless of the polarity of foreground-background shades. The method employs an edge-based connected component approach and automatically determines a threshold for each component. It has several advantages over existing binarization methods. Firstly, it can handle documents with multi-colored texts with different background shades. Secondly, the method is applicable to documents having text of widely varying sizes, usually not handled by local binarization methods. Thirdly, the method automatically computes the threshold for binarization and the logic for inverting the output from the image data and ***does not require any input parameter****. The proposed method has been applied to a broad domain of target document types and environment and is found to have a good adaptability.*

1 Introduction

There has been an increased use of cameras in acquiring document images as an alternative to traditional flatbed scanners and research towards camera based document analysis is growing [3]. Digital cameras are compact, easy to use, portable and offer a high-speed non-contact mechanism for image acquisition. The use of cameras has greatly eased document acquisition and has enabled human interaction with any type of document. Its ability to capture non-paper document images like scene text has several potential applications like licence plate recognition, road sign recognition, digital note taking, document archiving and wearable computing. But at the same time, it has also presented us with much more challenging images for any recognition task. Traditional scanner-based document analysis systems fail against this new and promising acquisition mode. Camera images suffer from uneven lighting, low resolution, blur, and perspective distortion. Overcoming these challenges will help us effortlessly acquire and manage information in documents.

In most document processing systems, a binarization process precedes the analysis and recognition procedures. The use of two-level information greatly reduces the computational load and the complexity of the analysis algorithms. It is critical to achieve robust binarization since any error introduced in this stage will affect the subsequent processing steps. The simplest and earliest method is the global thresholding technique that uses a single threshold to classify image pixels into foreground or background classes. Global thresholding techniques are generally based on histogram analysis [4, 6]. It works well for images with well separated foreground and background intensities. However, most of the document images do not meet this condition and hence the application of global thresholding methods is limited. Camera-captured images often exhibit non-uniform brightness because it is difficult to control the imaging environment unlike the case of the scanner. The histogram of such images are generally not bi-modal and a single threshold can never yield an accurate binary document image. As such, global binarization methods are not suitable for camera images. On the other hand, local methods use a dynamic threshold across the image according to the local information. These approaches are generally window-based and the local threshold for a pixel is computed from the gray values of the pixels within a window centred at that particular pixel. Niblack [5] proposed a binarization scheme where the threshold is derived from the local image statistics. The sample mean $\mu_{(x,y)}$ and the standard deviation $\sigma_{(x,y)}$ within a window W centred at the pixel location (*x,y*) are used to compute the threshold T$_{(x,y)}$ as follows:

$$T_{(x,y)} = \mu_{(x,y)} - k\,\sigma_{(x,y)}, \quad k = 0.2 \tag{1}$$

Yanowitz and Bruckstein [10] introduced a threshold that varies over different image regions so as to fit the spatially changing background and lighting conditions. Based on the observation that the location and gray level values at the edge points of the image are good choices for local thresholds, a threshold surface is created by relaxation initialized on the edge points. The method is however computationally very intensive. Trier and Jain [8] evaluated 11 popular local thresholding methods on scanned documents and reported that Niblack's method performs the best for optical character recognition (OCR). The method works well if the window encloses at least 1-2 characters. However, in homogeneous regions larger than size of the window, the method produces a noisy output since the expected sample variance becomes the background noise variance. Sauvola and Pietikainen [7] proposed an improved version of the Niblack's method by introducing a hypothesis that the gray values of the text are close to 0 (Black) while the background pixels are close to 255 (White). The threshold is computed with the dynamic range of standard deviation (R) which has the effect of amplifying the contribution of standard deviation in an adaptive manner.

$$T_{(x,y)} = \mu_{(x,y)}\,[1 + k\,(\frac{\sigma_{(x,y)}}{R} - 1)] \qquad (2)$$

where the parameters R and k are set to 128 and 0.5 respectively. This method minimizes the effect of background noise and is more suitable for document images. As pointed out by Wolf *et al* in [9], the Sauvola method fails for images where the assumed hypothesis is not met and accordingly, they proposed an improved threshold estimate by taking the local contrast measure into account.

$$T_{(x,y)} = (1-a)\mu_{(x,y)} + a\mathrm{M} + a\frac{\sigma_{(x,y)}}{\mathrm{S}_{max}}(\mu_{(x,y)} - \mathrm{M}) \qquad (3)$$

where M is the minimum value of the grey levels of the whole image, S_{max} is the maximum value of the standard deviations of all windows of the image and 'a' is a parameter fixed at 0.5. The Wolf's method requires two passes since one of the threshold decision parameter S_{max} is the maximum of all standard deviation of all windows of the images. The computational complexity is therefore slightly higher in this case. This method combines Savoula's robustness with respect to background textures and the segmentation quality of Niblack's method.

However, recent developments on document types, for example, documents with both graphics and text, where the text varies in color and size, call for more specialized binarization techniques. It is relatively difficult to obtain satisfactory binarization with various kinds of document images. The choice of window size in local methods can severely affect the result of binarization and may give rise to broken characters and voids, if the characters are thicker than the size of the window considered. Moreover, we often encounter text of different colors in a document image. Conventional methods assume that the polarity of the foreground-background intensity is known a priori. The text is generally assumed to be either bright on a dark background or vice versa. If the polarity of the foreground-background intensity is not known, the binary decision logic could treat some text as background and no further processing can be done on those text. Clark and Mirmhedi [2] use a simple decision logic to invert the result of binarization based on the assumption that the background pixels far outnumber the text pixels. Within each window, the number of pixels having intensity values higher or lower than the threshold are counted and the one which is less in number is treated as the foreground text. This simple inversion logic cannot handle the case where the characters are thick and occupy a significant area of the window under consideration. Moreover, a document image can have two or more different shades of text with different background colors as shown in Fig. 1. Binarization using a single threshold on such images, without a priori information of the polarity of foreground-background intensities, will lead to loss of textual information as some of the text may be assigned as background. The characters once lost cannot be retrieved back and are not available for further processing. Possible solutions need to be sought to overcome this drawback so that any type of document could be properly binarized without the loss of textual information.

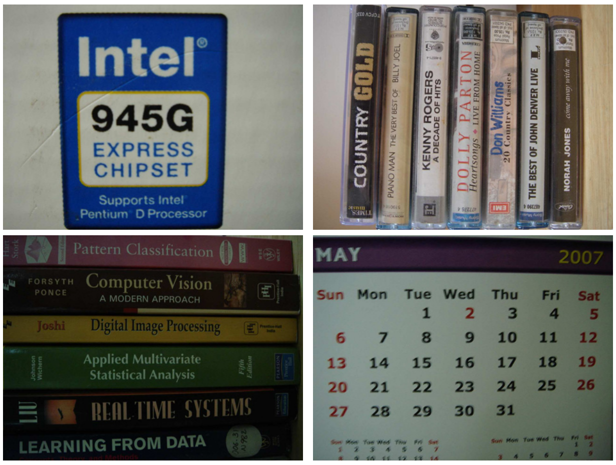

Intel
945G
EXPRESS CHIPSET
Supports Intel Pentium D Processor
COUNTRY GOLD
KENNY ROGERS A DECADE OF HITS
DOLLY PARTON
Don Williams 20 Country Classics
THE BEST OF JOHN DENVER LIVE
NORAH JONES
Pattern Classification
Computer Vision A MODERN APPROACH
Digital Image Processing
Applied Multivariate Statistical Analysis
REAL TIME SYSTEMS
LEARNING FROM DATA
MAY 2007
Sun Mon Tue Wed Thu Fri Sat

Figure 1. Some example images with multicolored textual content and varying background shades. A conventional binarization technique, using a fixed foreground-background polarity, will treat some characters as background, leading to the loss of some textual information

2 System Description

Text is the most important information in a document. We propose a novel method to binarize camera-captured color document images, whereby the foreground text is output as black and the background as white irrespective of the original polarity of foreground-background shades. The proposed method uses an edge-based connected component approach to automatically obtain a threshold for each component. Canny edge detection [1] is performed individually on each channel of the color image and the edge map $\mathbf{E}$ is obtained by combining the three edge images as follows

$$\mathbf{E} = \mathbf{E}_R \vee \mathbf{E}_G \vee \mathbf{E}_B \tag{4}$$

Here, $\mathbf{E}_R$, $\mathbf{E}_G$ and $\mathbf{E}_B$ are the edge images corresponding to the three color channels and $\vee$ denotes the logical OR operation. An 8-connected component labeling follows the edge detection step and the associated bounding box information is computed. We call each component, thus obtained, an edge-box (EB). We make some sensible assumptions about the document and use the area and the aspect ratios of the EBs to filter out the obvious non-text regions. The aspect ratio is constrained to lie between 0.1 and 10 to eliminate highly elongated regions. The size of the EB should be greater than 15 pixels but smaller than 1/5th of the image dimension to be considered for further processing.

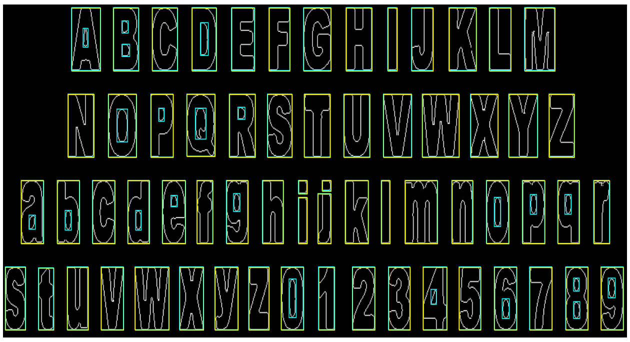


Figure 2. Edge-boxes for the English alphabet and numerals. Note that there is no character that completely encloses more than two edge components

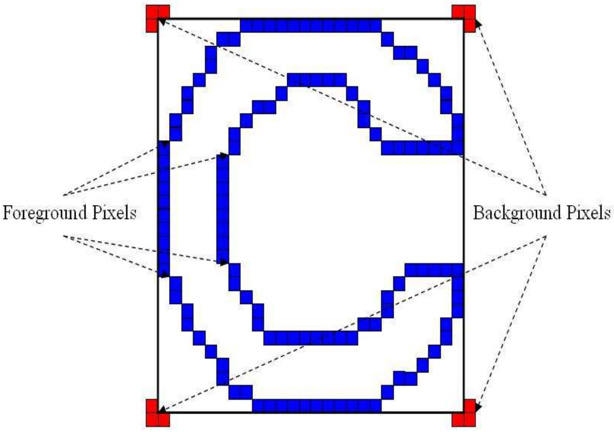


Figure 3. The foreground and the background pixels of each edge component

Since the edge detection captures both the inner and outer boundaries of the characters, it is possible that an EB may completely enclose one or more EBs as illustrated in Fig. 2. For example, the letter ‘O’ gives rise to two components; one due to the inner boundary EB$_{int}$ and the other due to the outer boundary EB$_{out}$. If a particular EB has exactly one or two EBs that lie completely inside it, the internal EBs can be conveniently ignored as it corresponds to the inner boundaries of the text characters. On the other hand, if it completely encloses three or more EBs, only the internal EBs are retained while the outer EB is removed as such a component does not represent a text character. Thus, the unwanted components are filtered out by subjecting each edge component to the following constraints:

```
if ($N_{int} < 3$)
  {Reject $EB_{int}$, Accept $EB_{out}$}
else
  {Reject $EB_{out}$, Accept $EB_{int}$}
```

where EB$_{int}$ denotes the EBs that lie completely inside the current EB under consideration and N$_{int}$ is the number of EB$_{int}$. These constraints on the edge components effectively remove the obvious non-text elements while retaining all the text-like elements. Only the filtered set of EBs are considered for binarization.

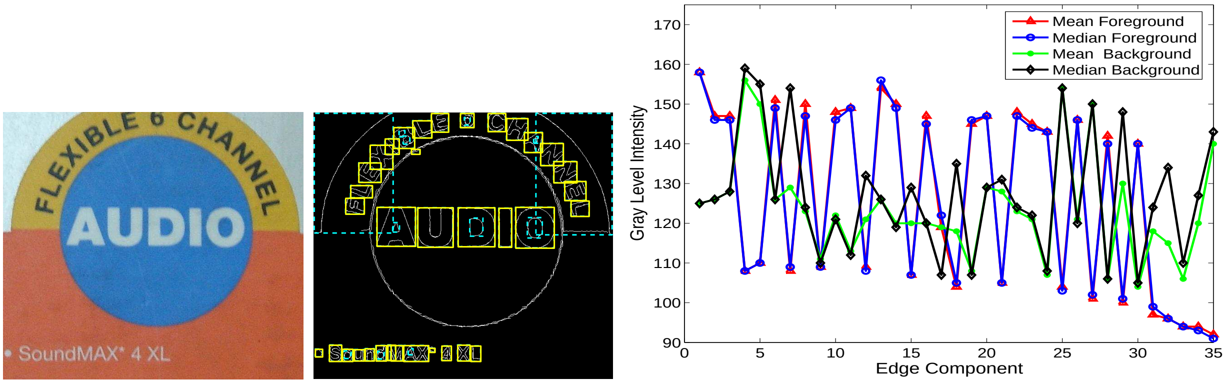
3 Estimation of Threshold

For each EB, we estimate the foreground and background intensities and the threshold is computed individually. Fig. 3 shows the foreground and the background pixels which are used for obtaining the threshold and inversion of the binary output.

The foreground intensity is computed as the mean gray-level intensity of the pixels that correspond to the edge pixels.

$$F_{EB} = \frac{1}{\mathrm{N}_E} \sum_{(x,y)\in\mathbf{E}} \mathbf{I}(x,y) \tag{5}$$

where $\mathbf{E}$ represent the edge pixels, $\mathbf{I}(x,y)$ represent the in-



(a) (b) (c)

Figure 4. (a) Input Image (b) Output of Edge-Box filtering. The dotted boxes in cyan are filtered out and only the yellow solid boxes (35 in number) are considered for binarization (c) The threshold parameters for the valid edge components. Observe that the mean and median intensities of the foreground pixels are almost the same for all characters. The same holds true for the background estimate for horizontally (or vertically) aligned text. However, when the text is aligned diagonally, the mean intensity of the background pixels is affected due to overlapping of the adjacent bounding boxes. Hence, the median intensity gives a more reliable logic for inverting the binary output

tensity value at the pixel (*x,y*) and N$_E$ is the number of edge pixels in an edge component.

For obtaining the background intensity, we consider three pixels each at the periphery of the corners of the bounding box as follows

$$\mathbf{B} = \{\mathbf{I}(x-1,y-1),\mathbf{I}(x-1,y),\mathbf{I}(x,y-1),\\ \mathbf{I}(x+w+1,y-1),\mathbf{I}(x+w,y-1),\mathbf{I}(x+w+1,y),\\ \mathbf{I}(x-1,y+h+1),\mathbf{I}(x-1,y+h),\mathbf{I}(x,y+h+1),\\ \mathbf{I}(x+w+1,y+h+1),\mathbf{I}(x+w,y+h+1),\mathbf{I}(x+\\ w+1,y+h)\}$$

where (*x, y*) represent the coordinates of the top-left corner of the bounding-box of each edge component and w and h are its width and height, respectively. Fig. 4 shows the output of the edge-box filtering and the threshold parameters for each of the valid edge components. As it is observed in Fig. 4(c), the mean or median intensity are almost the same for the foreground pixels irrespective of the text orientation. However, for a diagonally aligned text, the bounding boxes can have some overlap with the adjacent components and can interfere in the background intensity estimate. This is the case for the text 'FLEXIBLE 6 CHANNEL' printed in black in a semi-circular manner which are represented in Fig. 4(c) by the edge components whose estimated foreground intensity is lower than that of the background. The mean background intensity for these components are affected by the adjacent components while the median is not. Thus, the local background intensity can be estimated more reliably by considering the median intensity of the 12 background pixels instead of the mean intensity.

$$B_{EB} = \text{median}(\mathbf{B}) \tag{6}$$

Assuming that each character is of uniform color, we binarize each edge component using the estimated foreground intensity as the threshold. Depending on whether the foreground intensity is higher or lower than that of the background, each binarized output $\mathbf{BW}_{EB}$ is suitably inverted so that the foreground text is always black and the background always white.

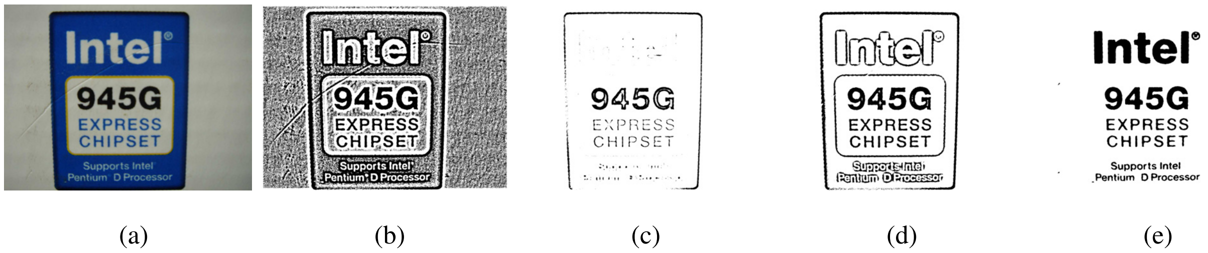
$$\text{If } F_{EB} < B_{EB},\ \mathbf{BW}_{EB}(x,y) = \begin{cases} 1, & \mathbf{I}(x,y) \geq F_{EB} \\ 0, & \mathbf{I}(x,y) < F_{EB} \end{cases} \tag{7}$$

$$\text{If } F_{EB} > B_{EB},\ \mathbf{BW}_{EB}(x,y) = \begin{cases} 0, & \mathbf{I}(x,y) \geq F_{EB} \\ 1, & \mathbf{I}(x,y) < F_{EB} \end{cases} \tag{8}$$

All the threshold parameters explained in this section are derived from the image data and the method is thus completely free from user-defined parameters.

4 Experiments

The test images used in this work are acquired from a Sony digital still camera at a resolution of 1280 × 960. The images are taken from both physical documents such



(a) (b) (c) (d) (e)

Figure 5. Comparison of some popular local binarization methods for (a) a document image with multiple text colors and sizes (b) Niblack's Method (c) Sauvola's Method and (d) Wolf's Method with the (e) Proposed method. While the proposed method is able to handle characters of any size and color, all other methods fail to binarize properly the components larger than the size of the window (25 × 25 used here) and require a priori knowledge of the polarity of foreground-background intensities as well

.

as book covers, newspapers etc and non-paper document images like text on 3-D real world objects. The connected component analysis performed on the edge map captures all the text characters irrespective of the polarity of their foreground and background intensities. We have used the thresholds 0.2 and 0.3 for the hysteresis thresholding step of Canny edge detection. The variance of the associated Gaussian function is taken to be 1. The constraints on the edge components effectively removes the obvious non-text elements while retaining all the text-like components. From each valid edge component, the foreground and background intensities are automatically computed and each of them is binarized individually.

Fig. 5 compares the results of our method with some popular local binarization techniques, namely, Niblack's method, Sauvola's method and Wolf's method on a document image having multi-colored text and large variations in sizes with the smallest and the largest components being 7×5 to 291×174 respectively. Clearly, these local binarization methods fail when the size of the window is smaller than stroke width. A large character is broken up into several components and undesirable voids occur within thick characters. It requires a priori knowledge of the polarity of foreground-background intensities as well. On the other hand, our method can deal with characters of any size and color as it only uses edge connectedness.

The binarization logic developed here is tested on documents having foreground text with different background shades. Though these kinds of images are quite commonly encountered, none of the existing binarization techniques can deal with such images. The generality of the algorithm is tested on more than 50 complex color document images and is found to have a high adaptivity and performance. Some results of binarization using our method are shown in Fig. 6 adjacent to its respective input images. The algorithm deals only with the textual information and it does not threshold the edge components that were already filtered out. In the resulting binary images, as desired, all the text regions are output as black while the background as white, irrespective of their colors in the input images.

5 Conclusions and Future Work

We have developed a novel technique for binarization of text from digital camera images. It has a good adaptability without the need for manual tuning and can be applied to a broad domain of target document types and environment. It simultaneously handles the ambiguity of polarity of the foreground-background shades and the algorithm's dependency on the parameters. The edge-box analysis captures all the characters, irrespective of their sizes thereby enabling us to perform local binarization without the need to specify any window. The use of edge-box has enabled us to automatically compute the foreground and the background intensities reliably and hence the required threshold for binarization. The proposed method retains the useful textual information more accurately and thus, has a wider range of applications compared to other conventional methods.

The edge detection method is good in finding the character boundaries irrespective of the foreground-background polarity. However, if the background is textured, the edge components may not be detected correctly due to edges from the background and our edge-box filtering strategy fails. This has been observed for the image shown in Fig. 7. Overcoming these challenges is considered as a future extension to this work. The method is able to capture all the text while at the same time, filter out most of the components due to the background. The method can be extended to incorporate text localization as well.

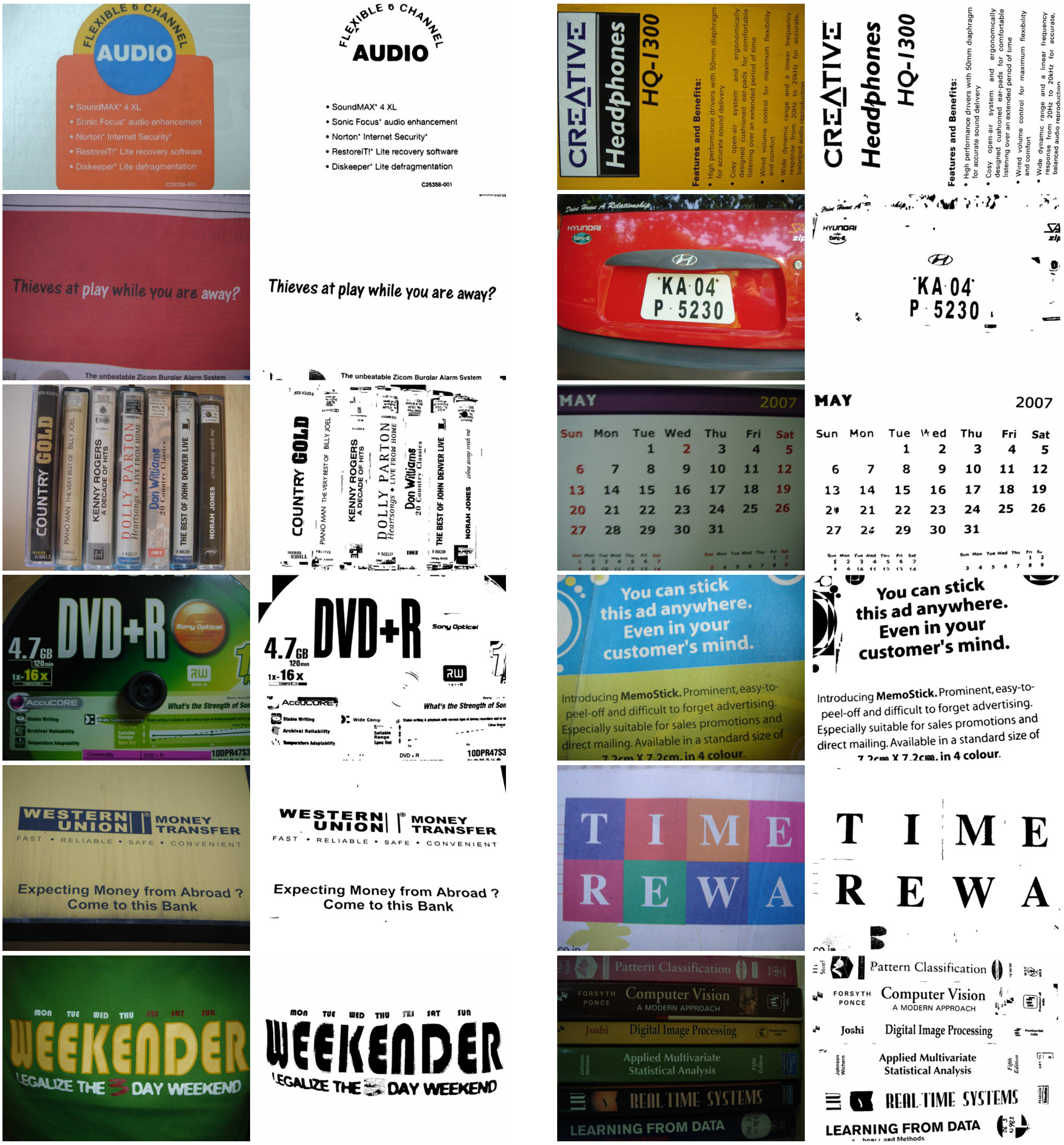

FLEXIBLE 6 CHANNEL
AUDIO
• SoundMAX* 4 XL
• Sonic Focus* audio enhancement
• Norton* Internet Security*
• RestoreIT!* Lite recovery software
• Diskeeper* Lite defragmentation
C25358-001
CREATIVE
Headphones
HQ-1300
Features and Benefits:
• High performance drivers with 50mm diaphragm for accurate sound delivery
• Cosy open-air system and ergonomically designed cushioned ear-pads for comfortable listening over an extended period of time
• Wired volume control for maximum flexibility and comfort
• Wide dynamic range and a linear frequency response from 20Hz to 20kHz for accurate, balanced audio reproduction
Thieves at play while you are away?
The unbeatable Zicom Burglar Alarm System
HYUNDAI
KA 04
P 5230
COUNTRY GOLD
PIANO MAN THE VERY BEST OF BILLY JOEL
KENNY ROGERS A DECADE OF HITS
DOLLY PARTON Heartsongs • LIVE FROM HOME
Don Williams 20 Country Classics
THE BEST OF JOHN DENVER LIVE
NORAH JONES come away with me
MAY
2007
Sun Mon Tue Wed Thu Fri Sat
4.7GB
DVD+R
Sony Optical
1x-16x
AccuCORE
What's the Strength of Sony
10DPR47S3
You can stick this ad anywhere. Even in your customer's mind.
Introducing MemoStick. Prominent, easy-to-peel-off and difficult to forget advertising. Especially suitable for sales promotions and direct mailing. Available in a standard size of 7.2cm X 7.2cm, in 4 colour.
WESTERN UNION MONEY TRANSFER
FAST • RELIABLE • SAFE • CONVENIENT
Expecting Money from Abroad ? Come to this Bank
T I M E
R E W A
MON TUE WED THU FRI SAT SUN
WEEKENDER
LEGALIZE THE 3 DAY WEEKEND
Pattern Classification
FORSYTH PONCE
Computer Vision
A MODERN APPROACH
Joshi
Digital Image Processing
Applied Multivariate Statistical Analysis
REAL TIME SYSTEMS
LEARNING FROM DATA

Figure 6. Some examples of binarization results obtained using the proposed method. Based on the estimated foreground and background intensities, each binarized component is suitably inverted so that all the text are represented in black and the background in white

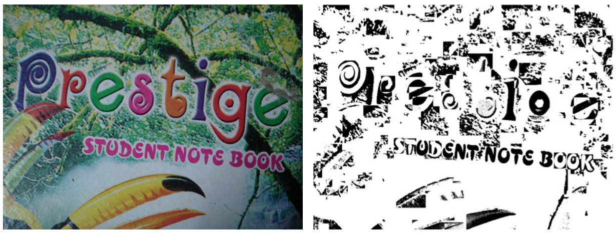


Figure 7. An example image for which the proposed algorithm fail to binarize properly

6 Acknowledgements

The authors wish to thank the anonymous reviewers for their useful comments and suggestions. The author, T. Kasar, would also like to thank his colleague A. Bhavna for the useful discussions on several aspects of this work.

References

[1] J. Canny. A computational approach to edge detection. *IEEE trans. PAMI*, 8(6):679–698, 1986.

[2] P. Clark and M. Mirmhedi. Rectifying perspective views of text in 3-d scenes using vanishing points. *Pattern Recognition*, 36:2673–2686, 2003.

[3] D. Doermann, J. Liang, and H. Li. Progress in camera-based document image analysis. *ICDAR*, 1:606–615, 2003.

[4] J. N. Kapur, P. K. Sahoo, and A. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision Graphics Image Process.*, 29:273–285, 1985.

[5] W. Niblack. An introduction to digital image processing. *Prentice Hall*, pages 115–116, 1986.

[6] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems Man Cybernetics*, 9(1):62–66, 1979.

[7] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.

[8] O. D. Trier and A. Jain. Goal-directed evaluation of binarization methods. *IEEE Trans. PAMI*, 17(12):1191–1201, 1995.

[9] C. Wolf, J. Jolion, and F. Chassaing. Text localization, enhancement and binarization in multimedia documents. *ICPR*, 4:1037–1040, 2002.

[10] S. Yanowitz and A. Bruckstein. A new method for image segmentation. *Computer Vision, Graphics and Image Processing*, 46(1):82–95, 1989.

A Framework Towards Realtime Detection and Tracking of Text

Carlos Merino
Departamento de Fisiología
Universidad de La Laguna
38071 Santa Cruz de Tenerife, Spain
cmerino@ull.es

Majid Mirmehdi
Department of Computer Science
University of Bristol
Bristol BS8 1UB, England
majid@cs.bris.ac.uk

Abstract

We present a near realtime text tracking system capable of detecting and tracking text on outdoor shop signs or indoor notices, at rates of up to 15 frames per second (for generous 640×480 *images), depending on scene complexity. The method is based on extracting text regions using a novel tree-based connected component filtering approach, combined with the Eigen-Transform texture descriptor. The method can efficiently handle dark and light text on light and dark backgrounds. Particle filter tracking is then used to follow the text, including SIFT matching to maintain region identity in the face of multiple regions of interest, fast displacements, and erratic motions.*

1. Introduction

Tracking text is an important step towards the identification and recognition of text for outdoor and indoor wearable or handheld camera applications. In such scenarios, as the text is tracked, it can be sent to OCR or to a text-to-speech engine for recognition and transition into digital form. This is beneficial in many application areas, such as an aid to the visually impaired or for language translation for tourists. Furthermore, the ability to automatically detect and track text in realtime is of use in localisation and mapping for human and robot navigation and guidance.

A review [9] and some collections of recent works [2, 1] in camera-based document analysis and recognition, highlight substantial progress in both single image and multi-frame based text analysis. Overall, there have been relatively few works on general text tracking. Multiframe text analysis has been mainly concerned with improving the text in a super-resolution sense [12] or for continuous recognition of text within a stationary scene e.g. on whiteboards or in slideshows [18, 20].

A directly related work in the area of general scene text tracking is by Myers and Burns [13] which successfully tracks scene text undergoing scale changes and 3D motion. However, this work applies to tracking in batch form and is not a realtime solution. Also in [13], the text detection is done by hand, manually indicating a starting bounding box for the tracking system to follow. Another work of interest is Li *et al.*[8] in which a translational motion tracking model was presented for caption text, based on correlation of image blocks and contour based stabilisation to refine the matched position. Less directly related, in [16], seven simple specific text strings were looked for by a roving camera from a collection of 55 images in an application to read door signs.

The focus of this paper is on the development of a resilient text tracking framework, using a handheld or wearable camera, as a precursor for our future work on text recognition. The only assumption we make is that we are looking for larger text sizes on shop and street signs, or indoor office boards or desktop documents, or other similar surfaces. Our proposed method is composed of two main stages: candidate text region detection and text region tracking. In the first stage, regions of text are located using a connected components approach combined with a *texture* measure step [17] which to the best of our knowledge has never been applied to text detection; this provides candidate regions or components which are then grouped to form possible words. The method is highly accurate but not infallible to noise, however, noisy or non-text candidate regions are not detected as persistently as true text regions, and can be rejected forthright during the tracking step. In the second stage, particle filtering is applied to track the text frame by frame. Each hypothesised system state is represented by a particle. The particles are weighted to represent the degree of belief on the particle representing the actual state. This non-linear filtering approach allows very robust tracking in the face of camera instability and even vigorous shakes. SIFT matching is used to identify regions from one frame to the next. We describe the details of this framework in the next few sections.

2. Background

It should be noted that there is a significant body of work on detecting (graphical) text that has been superimposed in images and videos, as well as in tracking such text. Example works are [10, 8]. In this work we concentrate solely on text embedded in natural scenes.

Segmentation of text regions involves the detection of text and then its extraction given the viewpoint. For example, almost each one of the works published in [2, 1] present one method or another for text segmentation, usually from a fronto-parallel point of view. Example past works considering other viewpoints and recovering the projective views of the text are [4, 14, 13]. Although in this work we engage in both *segmenting and tracking* text involving varying viewpoints, actual fronto-parallel recovery is not attempted. This is a natural step possible from the tracking motion information available and will be a key focus of our future work.

An issue of note is the problem of scale. Myers and Burns [13] dealt with this by computing homographies of planar regions that contain text, and when computationally tractable, this could be useful for any (realtime) text tracking application. Here, we are detecting text dynamically, hence at some smaller scales our detector will simply not find it, until upon approach it becomes large enough.

3. Methodology

The text tracking framework proposed here is based around the principle of a *tracker* representing a *text entity* - a word or group of words that appear together in an image as a salient feature, where each word comprises two or more components or regions. Trackers are dynamically created when a new text entity is detected; they follow the text frame to frame, and they get removed when the text cannot be detected anymore. Partial occlusion is dealt with, and in cases of full occlusion, a new tracker starts once the text is back in view. Our text tracking framework involves text segmentation, text region grouping, and tracking, including dynamic creation and removal of trackers.

3.1. Text segmentation

The text segmentation stage uses a combination of a connected components (CC) approach and a region filtering stage, with the latter involving the novel application to text analysis of a *texture* measure. The resulting component regions are then grouped into text entities.

3.1.1 Connected component labelling Following CC labelling in [7], León *et al* employed a tree pruning approach to detect text regions. They thresholded the image at every grey level, and built a Max-tree representation where each node stored the CC of the corresponding threshold level. The leaves of the tree represented the zones whose grey levels were the highest in the image. For detection of dark text over bright backgrounds, they built a different tree, a Min-tree, where the leaves represented the zones with the lowest grey levels in the image. This two pass treatment of bright text and dark text is very common in text detection algorithms.

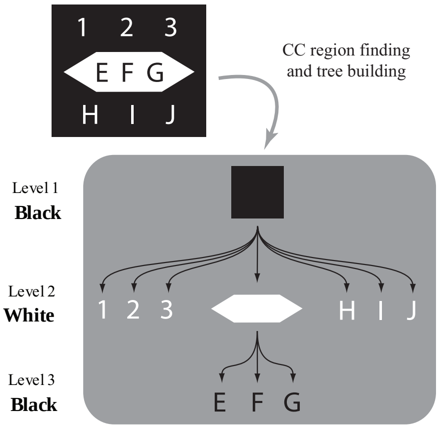


Figure 1. A synthetic sample image and its corresponding tree of connected regions.

We improve on the tree region labelling method in [7] by introducing a simple representation that allows efficient, one pass detection of bright text (white over black) and dark text (black over white) in the same tree. Initially, simple local adaptive thresholding is applied to the source frame. We empirically fixed the local threshold window size to 17×17 throughout all our experiments. The threshold was the mean grey level value of the window itself. Connected component region labelling is then performed on the thresholded image. This labelling builds a tree of connected regions, with the outermost region the root of the tree and the innermost regions the leaves. We allow the regions to toggle their label value from black to white as we go down each level of the tree. The tree represents the nesting relationship between these regions. Each node of the tree keeps only the *contour* around the border of the regions (see Figure 1).

Once the tree is built, it is walked depth-first with the objective to filter out the regions that are not text. Each node of the tree is classified as text or non-text during the walk using region filtering as described later below.

Usually, on real-world images with scene text, structural elements (such as sign borders, posters frames, etc.) can exhibit characteristics of text, such as high contrast against their backgrounds or strong texture response. These elements can be easily discarded (as long as they are not at a leaf) using the nesting relationship present in the proposed tree. When a node has children already classified as text,

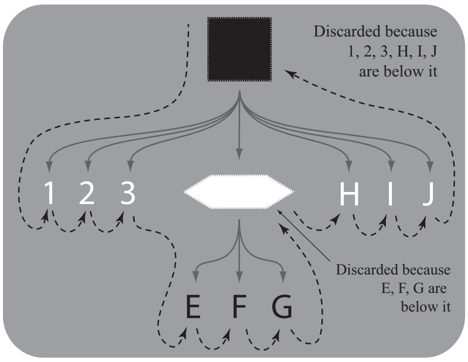


Figure 2. Parent nodes are discarded when children are classified as text.

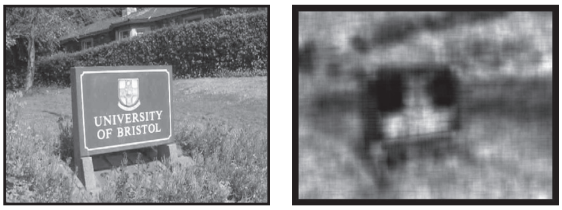


Figure 3. Original image and its Eigen-Transform response.

it is discarded as non-text, despite the text classifying functions may having marked it as text. This discards most of the non-text structural elements of the text (Figure 2).

3.1.2 Region filtering To classify text regions we apply three tests in cascade, meaning that if a test discards a region as non-text, no more tests are applied to it. This is in a similar fashion to Zhu *et al.* [21] who used 12 classifiers. In our case, the fewer tests are important for real time processing, and coarse, but computationally more efficient tests are applied first, quickly discarding obvious non-text regions, and slower, more discriminative tests are applied last, where the number of remaining regions is fewer. The test we apply are on *size*, *border energy*, and an eigenvector based *texture measure*.

Size - Regions too big or too small are discarded. The thresholds here are set to the very extreme. Very small regions are discarded to avoid noise. This may still drop characters, but they probably would be otherwise impossible to recognise by OCR and as the user gets closer, they are more likely to be picked up anyway. Large regions are discarded because it is unlikely that a single character occupies very large areas (over half the size) of the image.

Border energy - A Sobel edge operator is applied to all the points along the contour of each component region r and the mean value is obtained:

$$B_r = \frac{\sum_{i=1}^{P_r} \sqrt{(G_{ix}^2 + G_{iy}^2)}}{P_r} \tag{1}$$

where P_r denotes the number of border pixels in region r, and G_x and G_y represent the Sobel gradient magnitudes. This is referred to as the *border energy* and provides a measurement of region to background contrast. Regions with border energy value below a very conservatively fixed threshold are discarded. This removes regions that usually appear in less textured and more homogeneous regions.

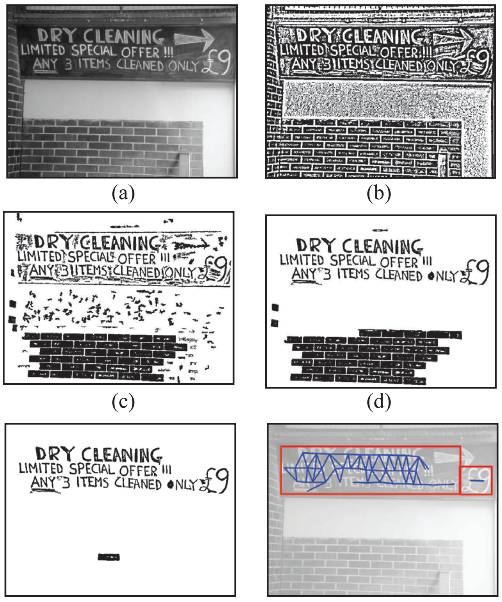
Jiang *et al* [6] used a three level Niblack threshold [19] in their text detection technique with good results. This introduces the local pixel values variance into the threshold calculation. However, this involves computing the standard deviation of local pixel values and we have found that doing a simpler adaptive threshold and afterwards discarding the noisy regions is faster. Also, the proposed tree walking algorithm transparently manages bright-text and dark-text occurrences on the same image without the need to apply a three level threshold image.

Texture measure - For this final decision-making step we apply a texture filter whose response at positions within the region pixels and their neighbourhoods is of interest.

We have previously combined several texture measures to determine candidate text regions, see [3]. These were mainly tuned for small scale groupings of text in the form of paragraphs. Although quite robust, the need for faster processing precludes their combined use. Here, we introduce the use of the Eigen-Transform texture operator [17] for use in text detection. It is a descriptor which gives an indication of surface roughness. For a square $w \times w$ matrix representing a pixel and its neighbouring grey values, the eigenvalues of this matrix are computed: $\|\lambda_1\| \geq \|\lambda_2\| \geq \ldots \|\lambda_w\|$. The largest l eigenvalues are discarded since they encode the lower frequency information of the texture. Then, the Eigen-Transform of the central pixel is the mean value of the $w - l + 1$ smaller magnitude eigenvalues:

$$\Gamma(l, w) = \frac{1}{w - l + 1} \sum_{k=l}^{w} \|\lambda_k\| \tag{2}$$

The Eigen-Transform has a very good response to texture which exhibit high frequency changes, and we found in our experiments that it responds to text very well for this reason, see a simple example in Figure 3 where both the text and the background texture are picked up well. It can, however, be a fairly slow operator, but fortunately we need only apply it to the component region pixels. Indeed, we compute the Eigen-Transform only on some regularly sampled points inside the bounding box of each region of interest. A key factor in (2) is the size of w. This is determined automatically by setting it dynamically according to the height



(a) (b) (c) (d) (e) (f)

Figure 4. Steps of the text segmentation and grouping. (a) Original image, (b) Adaptive threshold, (c)–(e) result after filtering by size, border energy and Eigen-transform measure, (f) perceptual grouping.

of the region under consideration. Then l is set to be a fraction of w.

The result of the text segmentation stage is a set of candidate regions with a high likelihood of being text. For each region, the centre position of its bounding box is stored as a component $\mathbf{c}_i$ into the *observation function* $\mathbf{y}_k$ of the particle filter (see section 3.2). As a result of the CC region tree design, and taking into account only the contour and not the contents, both inverted text (light on dark) and normal text (dark on light) are detected in the same depth-first pass. Figure 4 shows an example result highlighting each of the text segmentation and grouping steps.

3.1.3 Perceptual text grouping - The text grouping stage takes the regions produced by the text segmentation step and makes compact groups of perceptually close or *salient* regions. We follow the work by Pilu [14] on perceptual organization of text lines for deskewing. Briefly, Pilu defines two scale-invariant saliency measures between two candidate text regions A and B: *Relative Minimum Distance* λ and *Blob Dimension Ratio* γ:

$$\lambda(A,B) = \frac{D_{\min}}{A_{\min} + B_{\min}} \qquad \gamma(A,B) = \frac{A_{\min} + A_{\max}}{B_{\min} + B_{\max}} \tag{3}$$

where $D_{\min}$ is the minimum distance between the two regions, and $A_{\min}$, $B_{\min}$, $A_{\max}$ and $B_{\max}$ are respectively the minimum and maximum axes of the regions A and B. Pilu's text saliency operator between two text regions is then:

$$\mathbf{P}(A,B) = N(\lambda(A,B),1,2) \cdot N(\gamma(A,B),0,4) \tag{4}$$

where $N(x,\mu,\sigma)$ is a Gaussian distribution with mean μ and standard deviation σ whose parameters were determined experimentally in [14]. To reduce the complexity of comparing all the regions against each other, we construct a planar graph using Delaunay triangulation, with the region centres as vertices. The saliency operator is then applied to each edge of this graph, keeping only the salient ones and removing the rest. This edge pruning on the graph effectively divides the original graph into a set of connected subgraphs. Each subgraph with more than two vertices is considered a text group. This additional filtering step removes a number of isolated regions (see Figure 4(f)).

3.2. Text tracking

Particle filtering, also known as the Sequential Monte Carlo Method, is a non-linear filtering technique that recursively estimates a system's state based on the available observation. In an optimal Bayesian context, this means estimating the *likelihood* of a system's state given the observation $p(\mathbf{x}_k|\mathbf{y}_k)$, where $\mathbf{x}_k$ is the system's state at frame k and $\mathbf{y}_k = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ is the observation function.

Each hypothesised new system state at frame k is represented by a particle resulting in $\{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(N)}\}$, where N is the number of particles. Each particle $\mathbf{x}_k^{(n)}$ has an associated weight $\left\{(\mathbf{x}_k^{(1)}, w_k^{(1)}), \ldots, (\mathbf{x}_k^{(N)}, w_k^{(N)})\right\}$ where $\sum_{i=1}^{s} w_k^{(i)} = 1$. Given the particle hypothesis $\mathbf{x}_k^{(n)}$, the weights are proportional to the likelihood of the observation, $p(\mathbf{y}_k|\mathbf{x}_k^{(n)})$. For a detailed explanation of particle filter algorithms and applications, see e.g. [5].

Particle filtering is the ideal method given the instability of the handheld or wearable camera in our application domain. We build on the particle tracking framework developed in [15] for simultaneous localisation and mapping (SLAM). Here we want to independently track multiple instances of text in the image, with a simple state representation. Thus, each text entity is assigned a particle filter, i.e. a *tracker*, responsible of keeping its state. The main components to now deal with in a particle filter implementation are the *state representation*, the *dynamics model* and the *observation model*.

3.2.1 State representation - The *tracker* represents the evolution over time of a text entity. It has a state that tries to model the apparent possible changes that the text entity may experience in the image context. The model has to be rich enough to approximate the possible transformations of the text but at the same time simple enough to be possible to estimate it in real time.

The state of a tracker at frame k is represented by a 2D translation and rotation: $\mathbf{x}_k = (t_x, t_y, \alpha)$. We found this simple state model provides sufficient accuracy given the degree of movement within consecutive frames, but is also important in computational savings towards a real-time model[1]. This state defines a relative coordinate space, where the x-axis is rotated by an angle α with respect to the image, and its origin is at (t_x, t_y) in image coordinates.

Let's say a text entity contains M components. Its tracker preserves a list of M features $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$ where each feature $\mathbf{z}_i$ is a 2D position lying in the tracker's relative coordinate space. Each feature represents a text component being tracked, and it is fixed during tracker initialization. We define the transformation function $\Psi(\mathbf{z}_i, \mathbf{x}_k)$ as the coordinate transform (translation and rotation) of a feature position from the state's coordinate space to image coordinates. This is used during weighting. Additionally, each feature is associated with a set of SIFT descriptors [11] computed only once during the tracker initialization. They give the ability to differentiate between candidate text components, providing a degree of multiscale and rotation invariance to the feature matching as well as resilience to noise and change in lighting conditions[2].

Figure 5 shows the current state representation $\mathbf{x}_k$ of a tracker at frame k which has $M = 4$ features $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4\}$. For ease of exposition, all the features are visualised to lie along the x-axis of the tracker's coordinate space. Further, the figure shows another particle $\mathbf{x}_k^{(1)}$ representing an alternative state hypothesis. The four features $\mathbf{z}_i \in \mathbf{Z}$ are mapped to the particle's relative coordinate space to show the same set of features from a different reference frame. The observation function $\mathbf{y}_k$, with $\mathbf{y}_k = \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$ representing the center points of the candidate text components is also shown.

3.2.2 Dynamics model - The dynamics model defines the likelihood of a system state transition between time steps as $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. It is composed of a deterministic part - a prediction of how the system will evolve in time, and a stochastic part - the random sampling of the particles around the predicted position. Examples of prediction schemes are constant position, constant velocity and constant acceleration. Examples of stochastic functions are uniform and Gaussian random walks around an uncertainty window of the predicted position.

The selection of an appropriate dynamics model is crucial for a tracking system to be able to survive unpredictable movements, such as those caused by wearable or hand-held camera movements. Pupilli [15] concluded that for such scenarios a constant position prediction model with a uniform or Gaussian random walk would provide better results, due to the unpredictable nature of erratic movements. Here, we follow this advice to use a constant position model with random Gaussian walk around the last state, i.e. $p(\mathbf{x}_k|\mathbf{x}_{k-1}) = N(\mathbf{x}_{k-1}, \Sigma)$. The covariance matrix Σ defines the *particle spread* which is empirically set to a generous size, and automatically reduced via an annealling process as in [15].

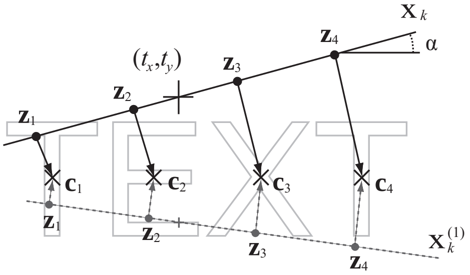


Figure 5. State model of one *tracker*, $\mathbf{x}_k = (t_x, t_y, \alpha)$, with 4 tracked features $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4\}$. A particle, $\mathbf{x}_k^{(1)}$, shows a different state hypothesis.

3.2.3 Observation model - Given a particle state hypothesis, the observation model defines the likelihood of the observation, $p(\mathbf{y}_k|\mathbf{x}_k^{(n)})$. The weight of each particle is calculated based on the comparison from projected features' positions and actual text components found in the image. An inlier/outlier likelihood proposed by Pupilli [15] is used.

For each tracked feature $\mathbf{z}_i \in \mathbf{Z}$, a set of candidate components $\mathbf{y}_{ki} \subseteq \mathbf{y}_k$ $\{(\mathbf{z}_1, \mathbf{y}_{k1}), (\mathbf{z}_2, \mathbf{y}_{k2}), \ldots, (\mathbf{z}_M, \mathbf{y}_{kM})\}$ is computed, based on their matching to the SIFT descriptors previously stored for each feature. This reduces the search space of the particles and gives robustness to the tracking process.

The weight of a particle is proportional to the number of observed candidate components inside a circular region of radius ε around each tracked feature. First an *inlier threshold* function $\tau(\mathbf{a}, \mathbf{b})$ is defined:

$$\tau(\mathbf{a}, \mathbf{b}) = \begin{cases} 1 & \text{if } d(\mathbf{a}, \mathbf{b}) < \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $d(\mathbf{a}, \mathbf{b})$ is the distance between two points. Then, the likelihood is:

$$p(\mathbf{y}_k|\mathbf{x}_k^{(n)}) \propto \exp\left(\sum_{\mathbf{z}_i \in \mathbf{Z}} \sum_{\mathbf{c}_j \in \mathbf{y}_{ki}} \tau\left(\Psi(\mathbf{z}_i, \mathbf{x}_k^{(n)}), \mathbf{c}_j\right)\right) \tag{6}$$

[1] However, we intend to investigate more complex motion models in future while ensuring the realtime aspects of the work are not compromised

[2] Note to Reviewers: We have found the SIFT matching to grossly slow our system down. By the time of this Workshop we will have implemented and hope to report faster invariant feature matching using e.g. the Hessian Affine or MSER which will additionally give a greater degree of affine invariancy

where $\Psi(\mathbf{z}_i, \mathbf{x}_k^{(n)})$ is the transformation function defined in subsection 3.2.1. Figure 6 shows the weighting process of one feature $\mathbf{z}_2$ for two different hypothesis, $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$. The latter is nearer to the actual state of the system and gets a greater weight. Note that for illustration purposes we are considering here that the candidate group components for feature $\mathbf{z}_2$ is all the observation: $\mathbf{y}_{k2} = \mathbf{y}_k = \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\}$.

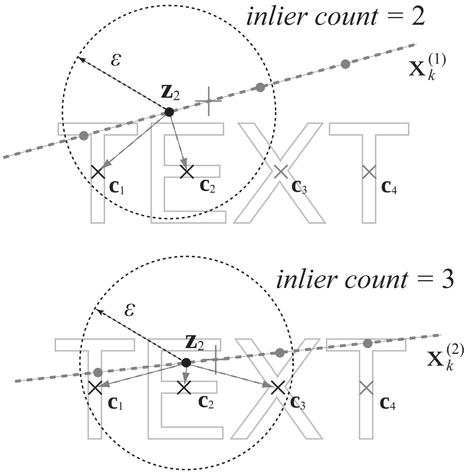


Figure 6. Inlier count of feature $\mathbf{z}_2$ for two different particles $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$.

3.2.4 Bounding box computation - Bounding box computation is crucial towards next possible stages such as extraction, recognition or superresolution. Thus, it is important that it is as stable and tight as possible. Once a posterior state is established by the particle filter, each feature $\mathbf{z}_i \in \mathbf{Z}$ is assigned a *Most Likely Estimate* (MLE), that is the text component $\mathbf{c}_j \in \mathbf{y}_k$ that most likely matches it. In Figure 5, the MLE of each feature is marked with an arrow. Not all tracked features will have a MLE each frame, as sometimes they are not found due to blur, clutter or occlusion.

After perceptual text grouping, each observed text component belongs to a group, and thus the MLE of each tracker feature also belongs to a group. The *Most Likely Group* (MLG) of a feature is the group to which this feature's MLE belongs to. Given this, the tracker's bounding box is then obtained by joining the bounding boxes of its MLGs.

3.2.5 Tracker creation and removal - Trackers are dynamically created when new text is detected, and removed when their associated text entity can no longer be found. After the grouping stage, any text group detected is a potential text entity to be tracked. But some of these groups may belong to text entities already being tracked. The tracking stage identifies the tracked components in the image via the MLE and MLG mechanisms. After the tracking cycle, any unidentified remaining groups are passed to a new tracker. Newly created trackers must continuously track their text for a number of frames to be considered *stable*. Trackers that fail to comply with this are promptly removed. The tracker removal mechanism is very simple. After a consecutive number of frames without a match, the track is considered lost and removed. Should the same text entity then reappear, it will be assigned a new tracker.

4. Results

The system was tested on a variety of typical outdoor and indoor scenarios, e.g. a hand-held camera while passing shops or approaching notices, posters, billboards etc. We present here the results from four typical scenarios. The full video sequences along with other results, including a sequence from [13], are also available online[3].

The results shown are: Figure 7: 'BORDERS' - walking in a busy street with several shop signs overhead, Figure 8: 'UOB' - walking past a signboard including an occlusion in a highly textured scene background, Figure 9'ST. MICHAEL'S HOSPITAL' - a traffic sign with both bright and dark text, complex background and significant perspective change, and Figure 10: 'LORRY' - with text also undergoing viewpoint changes. All sequences were at 640 × 480 resolution recorded at 15 fps with a consumer grade photo/video camera (Nikon Coolpix P2).

Table 1 shows the performance of the algorithm for the different sample scenes on an Intel Pentium IV 3.2Ghz processor. The results show the performance of the text segmentation and grouping subsystem alone, and the whole tracking process. Text segmentation is very fast. When measured off-line, the system was able to compute the results faster than the actual frame rate of the sequences. With the tracking, the performance of the system is *close to 10 fps on average*, depending on the complexity of the scene, making it promisingly close to realtime. For a simple scene with little background and one 5-character word, the *system could track it effortlessly at 15fps*. While the particle filtering framework is relatively fast, the SIFT matching of features reduces the performance when the number of candidate regions is large, such as in very complex backgrounds, e.g. in Fig. 8. A greater number of false positives (due to the vegetation) produced during segmentation put more stress on the tracking stage, which however rejected these regions due to the instability and lack of longevity of their trackers. Notice also in Fig. 8, the tracker survives the occlusion by the lamppost.

4.1. Discussion

The focus of this paper has been on a framework to track text as robustly and continuously as possible, bearing in

[3] Please see `http://vision.cs.bris.ac.uk/texttrack/`

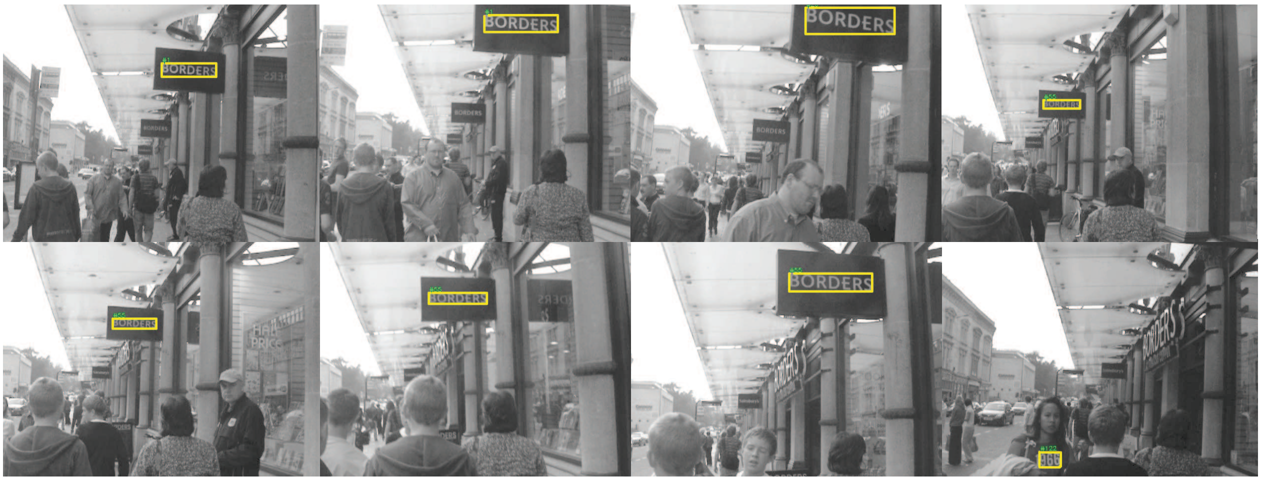


Figure 7. Example scene 1 - BORDERS - notice several BORDERS signs come along in the sequence.

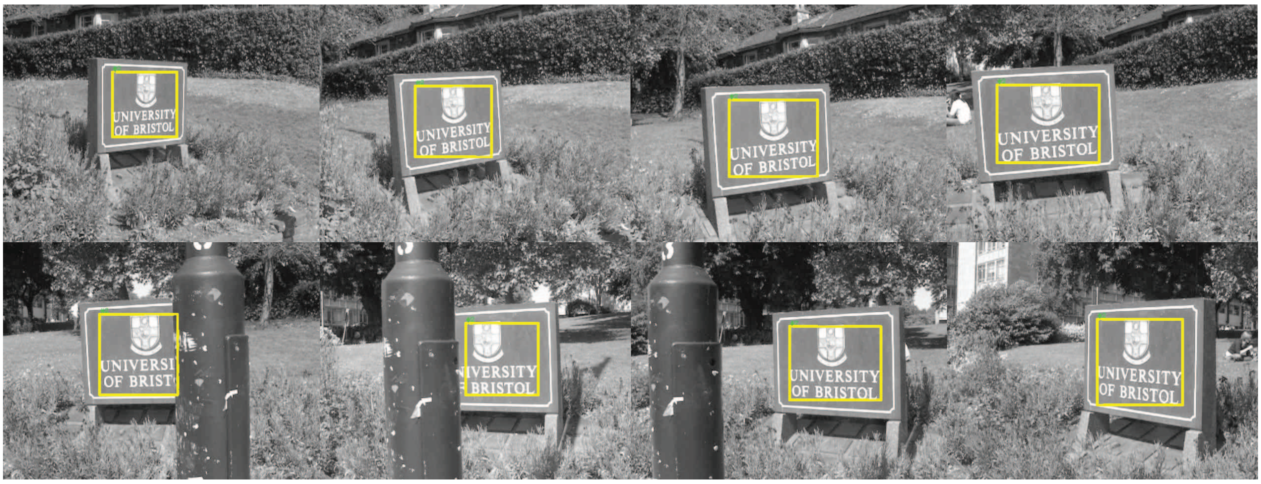


Figure 8. Example scene 2 - UOB including occlusion, also with much other texture.

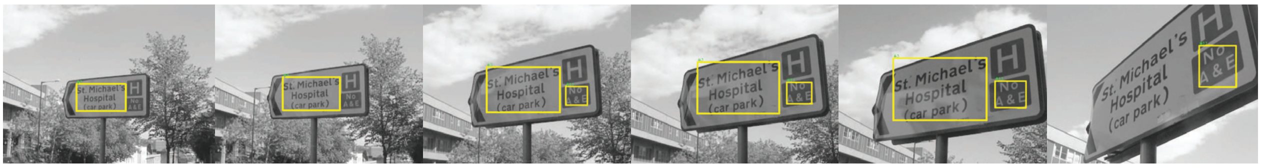


Figure 9. Example scene 3 - ST. MICHAEL’S HOSPITAL - two regions, dark over light and vice versa.

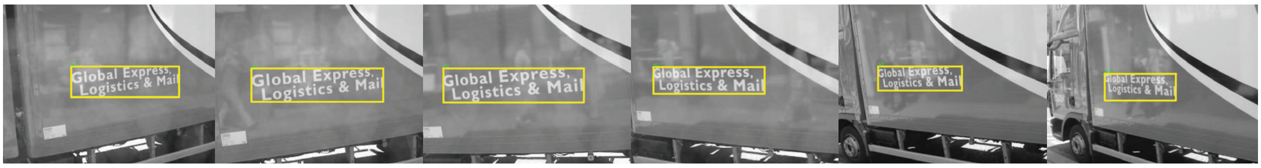


Figure 10. Example scene 4 - LORRY

Table 1. Performance of the algorithm in mean frames per second.

	Text segmentation	Full algorithm
Scene 1	31.9 fps	13.2 fps
Scene 2	21.3 fps	4.9 fps
Scene 3	30.7 fps	9.6 fps
Scene 4	32.0 fps	10.6 fps

mind that momentary loss of a text region is not disastrous in terms of recognition. Once stable tracking is obtained after a few frames, the motion information could be used for fronto-parallel recovery as well as generation of a super-resolution representation for better OCR, e.g. as in [12]. In our system, it is more likely that text is missed if it is at sharp perspective viewpoints, than for a non-text region to be tracked with significant stability. We had no such non-text cases, but even if there were, one can assume that OCR would reject it at the next stage.

Some shortcomings of our work are: (1) the robustness of our tracker improves further, in terms of dropping a track only to be picked up again instantly, when we use a more complex motion model, but this means we move further away from a realtime goal, (2) SIFT has limited robustness to viewpoint variations, so big changes of point of view will make the trackers lose the features, and it is by far the slowest part of the system, however we are at the time of writing experimenting with a new method, (3) Our results can not be claimed to be fully realtime, however we are near enough and believe we can achieve it in our future short-term work, (4) even though our few thresholds are fixed they naturally can affect the quality of the results; we aim to address these by applying learning techniques to automate the process where necessary.

5. Conclusion

In this paper we have presented a close to realtime technique to automatically detect and track text in arbitrary natural scenes. To detect the text regions, a depth-first search is applied to a tree representation of the image's connected components, where each leaf in the tree is examined for three criteria. Amongst these criteria is the use of the Eigen-Transform texture measure as an indicator of text. This stage of the algorithm detects both bright and dark text in a single traversal of the tree. Following perceptual grouping of the regions into text entities, particle filtering is applied to track them across sequences involving severe motions and shakes of the camera. We have established a significant framework and can start to improve its individual components in our future work to better our results.

Acknowledgements

Carlos Merino is funded by FIT-350300-2006-92 project from the Spanish Ministerio de Industria, Turismo y Comercio, through the European Regional Development Fund. This work was carried out partly at Bristol University and partly at Universidad de La Laguna and the Instituto Tecnológico de Canarias.

References

[1] *Proc. of the 1st Workshop on Camera Based Document Analysis and Recognition (CBDAR)*, August 2005.

[2] Special issue on camera-based text and document recognition. *International Journal on Document Analysis and Recognition*, 7(2–3), July 2005.

[3] P. Clark and M. Mirmehdi. Recognising text in real scenes. *IJDAR*, 4:243–257, 2002.

[4] P. Clark and M. Mirmehdi. Rectifying perspective views of text in 3d scenes using vanishing points. *Pattern Recognition*, 36(11):2673–2686, 2003.

[5] A. Doucet, J. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[6] R.-J. Jiang, F.-H. Qi, L. Xu, G.-R. Wu, and K.-H. Zhu. A learning-based method to detect and segment text from scene images. *Journal of Zhejiang University*, 8(4):568–574, 2007.

[7] M. León, S. Mallo, and A. Gasull. A tree structured-based caption text detection approach. In *Fifth IASTED VIIP*, 2005.

[8] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *IEEE-IP*, 9(1):147–156, 2000.

[9] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. *IJDAR*, 7(2):84–104, 2005.

[10] R. Lienhart. Indexing & retrieval of digital video sequences based on text recognition. In *ICM*, pages 419–420, 1996.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[12] C. Mancas-Thillou and M. Mirmehdi. Super-resolution text using the teager filter. In *CBDAR05*, pages 10–16, 2005.

[13] G. K. Myers and B. Burns. A robust method for tracking scene text in video imagery. In *CBDAR05*, 2005.

[14] M. Pilu. Extraction of illusory linear clues in perspectively skewed documents. In *CVPR*, pages 363–368, 2001.

[15] M. Pupilli. *Particle Filtering for Real-time Camera Localisation*. PhD thesis, University of Bristol, October 2006.

[16] H. Shiratori, H. Goto, and H. Kobayashi. An efficient text capture method for moving robots using dct feature and text tracking. In *ICPR*, pages 1050–1053, 2006.

[17] A. Targhi, E. Hayman, J. Eklundh, and M. Shahshahani. The eigen-transform & applications. In *ACCV*, pages I:70–79, 2006.

[18] M. Wienecke, G. A. Fink, and G. Sagerer. Toward automatic video-based whiteboard reading. *IJDAR*, 7(2):188–200, 2005.

[19] L. L. Winger, J. A. Robinson, and M. E. Jernigan. Low-complexity character extraction in low-contrast scene images. *IJPRAI*, 14(2):113–135, 2000.

[20] A. Zandifar, R. Duraiswami, and L. S. Davis. A video-based framework for the analysis of presentations/posters. *IJDAR*, 7(2):178–187, 2005.

[21] Z. Zhu, F. Qi, M. Kimachi, and Y. Wu. Using adaboost to detect & segment characters in natural scenes. In *CBDAR05*, 2005.

Section II

Retrieval

Camera Based Document Image Retrieval with More Time and Memory Efficient LLAH

Tomohiro Nakai, Koichi Kise, Masakazu Iwamura
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Naka, Sakai, Osaka, 599-8531 Japan
nakai@m.cs.osakafu-u.ac.jp, {kise, masa}@cs.osakafu-u.ac.jp

Abstract

In this paper, we propose improvements of our camera-based document image retrieval method with Locally Likely Arrangement Hashing (LLAH). While LLAH has high accuracy, efficiency and robustness, it requires a large amount of memory. It is also required to speed up the retrieval of LLAH for applications to real-time document image retrieval. For these purposes, we introduce the following two improvements. The first one is reduction of the required amount of memory by removing less important features for indexing from the database and simplifying structure of the database. The second improvement is to reduce exploring alternatives during the retrieval process. From the experimental results, we have confirmed that the proposed improvements realize reduction of the required amount of memory by about 80% and that of processing time by about 60%.

1. Introduction

Document image retrieval is a task of finding document images relevant to a given query from a database of a large number of document images. Various types of queries have been employed in document image retrieval [1]. Camera-based version of document image retrieval is characterized by its queries obtained by capturing documents with cameras. It has excellence that it enables linking paper documents to various services. In other words, paper documents can be viewed as media for providing services to the user. For example, the user can access relevant web sites by capturing a paper document when their URLs are related to the document images in the database.

We have already proposed a camera-based document image retrieval method based on a hashing technique called Locally Likely Arrangement Hashing (LLAH). LLAH is characterized by its accuracy, efficiency and robustness. It has been shown that more than 95% accuracy is realized with about 100 ms retrieval time on a 10,000 pages database[2]. Such accuracy and efficiency are realized by stable and discriminative features of LLAH. It has also been confirmed that LLAH is robust to *perspective distortion*, *occlusion* and *non-linear surface deformation* of pages which are typical problems for camera-captured documents [3, 4]. Features based on geometric invariants defined in local areas realize robustness to those problems.

In exchange for the accuracy and the robustness, LLAH requires a large amount of memory. For example, in order to realize accurate retrieval on a 10,000 pages database, 2.6GB memory is needed. Such heavy consumption of memory limits the scalability of LLAH. For the retrieval of 100,000 pages, for instance, the memory space is beyond what can be easily prepared. In addition, since real-time document image processing using cameras has significant usability [5], an application of LLAH to real-time document image retrieval is desired. In order for LLAH to be used in a real-time processing, further speeding up of its retrieval process is necessary.

In this paper, we propose some improvements of LLAH that solve the above problems. The basic idea of the improvement for the memory consumption is to remove unreliable features for indexing and simplify the structure of the database. As for the speeding up of processing, we introduce a feature-based method of reducing the number of combinations that need to be explored during the retrieval. From the experimental results using 10,000 document images, we have confirmed that the required amount of memory and the processing time are 1/5 and 2/5 of the original, respectively. We also show that the improved version of LLAH scales quite well: the memory consumption and the processing time is almost constant up to the database of size 10,000 images.

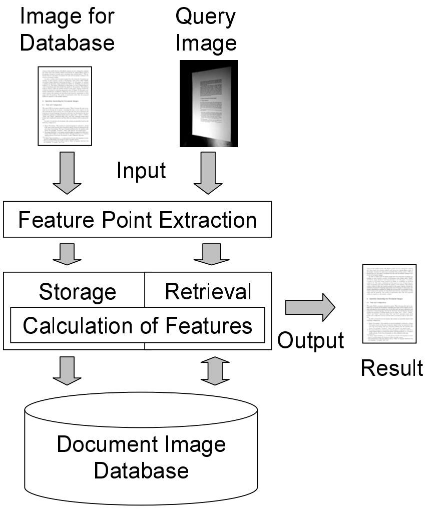


Figure 1. Overview of processing.

2. Document image retrieval with original LLAH

We first explain the original LLAH and the retrieval process with it.

2.1. Overview of processing

Figure 1 shows the overview of processing. At the step of feature point extraction, a document image is transformed into a set of feature points. Then the feature points are inputted into the storage step or the retrieval step. These steps share the step of calculation of features. In the storage step, every feature point in the image is stored independently into the document image database using its feature. In other words, a document image is indexed by using each feature point. In the retrieval step, the document image database is accessed with features to retrieve images by voting. We explain each step in the following.

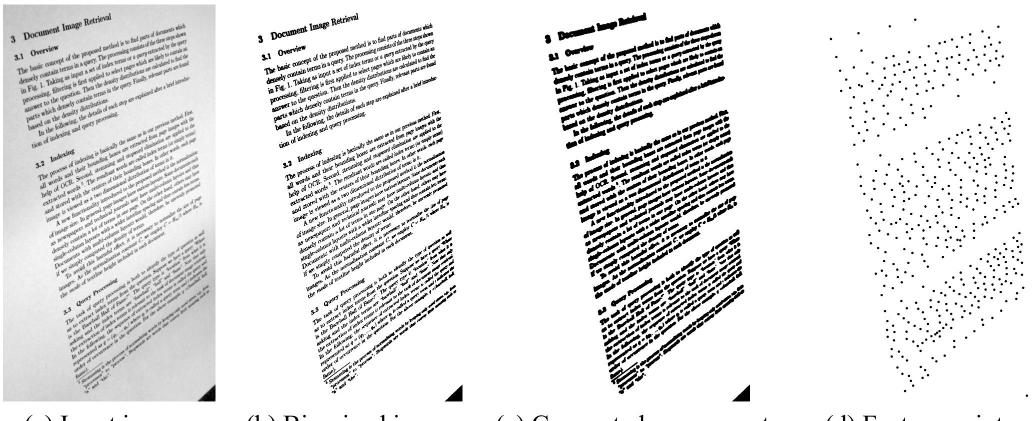
2.2. Feature point extraction

An important requirement of feature point extraction is that feature points should be obtained identically even under perspective distortion, noise, and low resolution. To satisfy this requirement, we employ centroids of word regions as feature points.

The processing is as follows. First, the input image (Fig. 2(a)) is adaptively thresholded into the binary image (Fig. 2(b)). Next, the binary image is blurred using the Gaussian filter. Then, the blurred image is adaptively thresholded again (Fig. 2(c)). Finally, centroids of word regions (Fig. 2(d)) are extracted as feature points.

2.3. Calculation of features

The feature is a value which represents a feature point of a document image. In order to realize successful retrieval, the feature should satisfy the following two requirements. One is that the same feature should be obtained from the same feature point even under various distortions. If different features are obtained from the same feature point at storage and retrieval processes, the corresponding document image cannot be retrieved. We call this requirement "stability of the feature". The other requirement is that different features should be obtained from different feature points. If the same feature is obtained from different feature points, not only the corresponding document image but also other document images are retrieved. We call this requirement "discrimination power of the feature". Both two require-



(a) Input image. (b) Binarized image. (c) Connected components. (d) Feature points.

Figure 2. Feature point extraction.

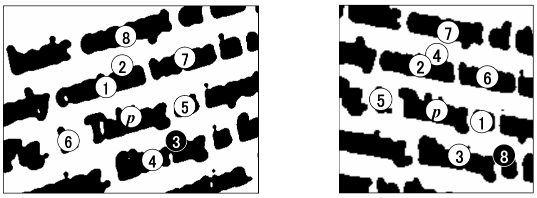


Figure 3. Eight nearest points of p in two images captured from different viewpoints. A circled number indicates its order of distance from p. Different points shown as black circles come up due to perspective distortion.

ments, the stability and the discrimination power, have to be satisfied for successful retrieval.

(1) Stability

In the cases of occlusion, the whole image of a document is not captured. In order to realize stability against occlusion, a feature has to be calculated from a part of document image. In LLAH, each feature point has its feature calculated from an arrangement of its neighboring points. Since features are calculated in a local part of a document image, the same feature can be obtained as long as the same part is captured.

Camera-captured images generally suffer from perspective distortion. In order to calculate features stable against perspective distortion, a geometric invariant is used. For this purpose, one may think the cross-ratio which is invariant to perspective transformation is appropriate. However, it is confirmed that an affine invariant gives higher accuracy than the cross-ratio [2]. This is because perspective transformation in a local area can be approximated as affine transformation and the affine invariant is more robust to change of feature points than the cross-ratio.

In this paper we utilize an affine invariant defined using four coplanar points ABCD as follows:

$$\frac{P(A,C,D)}{P(A,B,C)} \tag{1}$$

where P(A,B,C) is the area of a triangle with apexes A, B, and C.

The simplest definition of the feature of a feature point p is to use 4 nearest feature points from p. However, nearest feature points can change by the effect of perspective distortion as shown in Fig. 3. Hence the invariant from 4 nearest points is not stable. In order to solve this problem, we utilize feature points in a broader local area. In Fig. 3, it is shown that 6 points out of 7 nearest points are common. In general, we assume that common m points exist in n nearest neighbors under some extent of perspective distortion. Based on this assumption, we use common m points to calculate a stable feature. As shown in Fig. 4, common m points are obtained by examining all possible combinations $P_{m(0)}, P_{m(1)}, \cdots, P_{m({}_nC_m-1)}$ of m points from n nearest points. As long as the assumption holds, at least one combination of m points is common. Thus a stable feature can be obtained.

(2) Discrimination power

The simplest way of calculating the feature from m points is to set $m = 4$ and calculate the affine invariant from these 4 points. However, such a simple feature lacks the discrimination power because it is often the case that similar arrangements of 4 points are obtained from different feature points. In order to increase the discrimination power, we utilize again feature points of a broader area. It is performed by increasing the number $m(> 4)$. As m increases, the probability that different feature points have similar arrangement of m points decreases. As shown in Fig. 5, an arrangement of m points is described as a sequence of discretized invariants $(r_{(0)}, r_{(1)}, \cdots, r_{({}_mC_4-1)})$ calculated from all possible combinations of 4 feature points taken from m feature points.

2.4. Storage

Figure 6 shows the algorithm of storage of document images to the database. In this algorithm, the document ID is the identification number of a document, and the point ID is that of a point.

Next, the index H_{index} of the hash table is calculated by

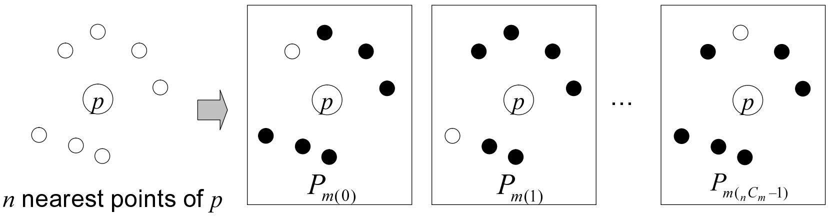


Figure 4. All possible combinations of $m(= 6)$ points from $n(= 7)$ nearest points are examined.

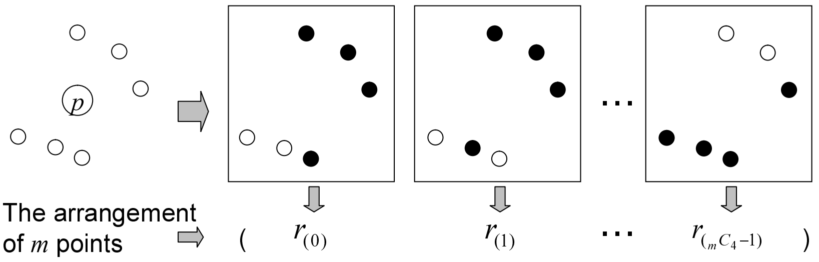


Figure 5. The arrangement of $m(=6)$ points is described as a sequence of invariants calculated from all possible combinations of 4 points.

```
for each $p \in$ {All feature points in a database image} do
    $P_n \leftarrow$ A set of the $n$ nearest points of $p$.
    for each $P_m \in$ { All combinations of $m$ points from $P_n$} do
        $L_m \leftarrow (p_0, \cdots, p_i, \cdots, p_{m-1})$ where $p_i$ is an ordered point of $P_m$ based on the angle from $p$ to $p_i$ with an arbitrary selected starting point $p_0$.
        $(L_4(0), \cdots, L_4(i), \cdots, L_4({}_mC_4 - 1)) \leftarrow$ A lexicographically ordered list of all possible $L_4(i)$ that is a subsequence consisting 4 points from $L_m$.
        for $i = 0$ to ${}_mC_4 - 1$ do
            $r_{(i)} \leftarrow$ a discretized affine invariant calculated from $L_4(i)$.
        end for
        $H_{\text{index}} \leftarrow$ The hash index calculated by Eq. (2).
        Store the item (document ID, point ID, $r_{(0)}$,···,$r_{({}_mC_4-1)}$) using $H_{\text{index}}$.
    end for
end for
```

Figure 6. Storage algorithm.

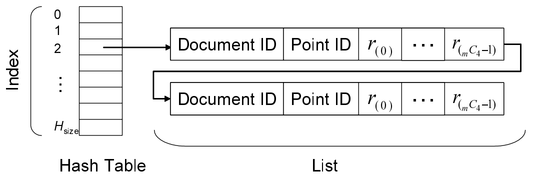


Figure 7. Configuration of the hash table.

```
1:  for each $p \in$ {All feature points in a query image} do
2:      $P_n \leftarrow$ A set of the $n$ nearest points of $p$.
3:      for each $P_m \in$ { All combinations of $m$ points from $P_n$} do
4:          for each $p_0 \in P_m$ do
5:              $L_m \leftarrow (p_0, \cdots, p_i, \cdots, p_{m-1})$ where $p_i$ is an ordered point of $P_m$ based on the angle from $p$ to $p_i$ with a starting point $p_0$.
6:              $(L_4(0), \cdots, L_4(i), \cdots, L_4({}_mC_4 - 1)) \leftarrow$ A lexicographically ordered list of all possible $L_4(i)$ that is a subsequence consisting 4 points from $L_m$.
7:              for $i = 0$ to ${}_mC_4 - 1$ do
8:                  $r_{(i)} \leftarrow$ a discretized affine invariant calculated from $L_4(i)$.
9:              end for
10:             $H_{\text{index}} \leftarrow$ The hash index calculated by Eq. (2).
11:             Look up the hash table using $H_{\text{index}}$ and obtain the list.
12:             for each item of the list do
13:                 if Conditions to prevent erroneous votes [2] are satisfied then
14:                     Vote for the document ID in the voting table.
15:                 end if
16:             end for
17:         end for
18:     end for
19: end for
20: Return the document image with the maximum votes.
```

Figure 8. Retrieval algorithm.

the following hash function:

$$H_{\text{index}} = \left(\sum_{i=0}^{{}_mC_4-1} r_{(i)} k^i \right) \bmod H_{\text{size}} \qquad (2)$$

where $r_{(i)}$ is a discrete value of the invariant, k is the level of quantization of the invariant, and H_{size} is the size of the hash table.

The item (document ID, point ID, $r_{(0)}, \cdots, r_{({}_mC_4-1)}$) is stored into the hash table as shown in Fig. 7 where chaining is employed for collision resolution.

2.5. Retrieval

The retrieval algorithm is shown in Fig. 8. In LLAH, retrieval results are determined by voting on documents represented as cells in the voting table.

First, the hash index is calculated at the lines 7 to 10 in the same way as in the storage step. At the line 11, the list shown in Fig. 7 is obtained by looking up the hash table. For each item of the list, a cell of the corresponding document ID in the voting table is incremented if it has the same feature $(r_{(0)},\cdots,r_{({}_mC_4-1)})$. Finally, the document which obtains the maximum votes is returned as the retrieval result.

3. Reduction of the required amount of memory

In this section, we introduce a method of reduction of the required amount of memory. In LLAH, many features are calculated in order to realize the stability of features. All features are stored in the hash table in the form of linked lists regardless to their importance. We reduce memory consumption by removing less important features and changing data structure of the database.

Let us show an example. In the following condition,

- $n = 7, m = 6, H_{\rm size} = 1.28 \times 10^8$
- The number of document images in the database is 10,000
- Average number of feature points in a document image is 630
- Document ID, point ID and $r_{(i)}$ are stored in 2 bytes, 2bytes and 1byte variables respectively
- A pointer variable requires 8 bytes

the hash table requires $1.28 \times 10^8 \times 8 = 1.0$GB and linked lists require $10,000\times630\times7\times(2+2+1\times15+8) = 1.2$GB. Therefore the total required amount of memory is 2.2GB.

Features which cause collisions in the hash table are less important because such features are shared by other points and thus likely to lack discrimination power. They also increase the processing time of retrieval since they frequently come up and increase the number of votes. From an experimental result, it is confirmed that the number of hash table entries with collisions is 28% of the number of hash table entries with at least one item. Since the number of entries with collisions is minor, removing features with collisions will not cause fatal effect on accuracy of retrieval. Thus we have decided to remove linked lists with collisions.

Removal of features with collisions enables further reduction of memory consumption since it allows to simplify the data structure of the hash table. Features (invariants $r_{(i)}$) are stored in order to find the appropriate item from the linked list in the case of collisions. Because we have eliminated all collisions, we can also remove the records of features. Moreover document IDs and point IDs are not needed to be stored in the form of linked list because only one item is stored at an index of the hash table. Therefore we adopt a simple hash table as shown in Fig. 9 as the structure of the database. Changing the data structure into a simple hash table enables further reduction of the memory consumption. For example, the required amount of memory is 512MB under the condition of the above example ($H_{\rm size} = 1.28\times10^8$, a document ID and a point ID are stored in 2 bytes variables). Therefore 77% reduction of the required amount of memory can be realized.

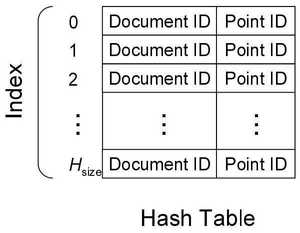


Figure 9. Configuration of the simplified hash table.

Another way to reduce the required amount of memory is to reduce the size of the hash table $H_{\rm size}$. However, $H_{\rm size}$ significantly affects performance of retrieval. Especially for the simplified hash table, the size of the hash table is fatal. If the hash table has an insufficient number of index space to store items, many entries will be invalidated due to collisions.

4. Speeding up retrieval

We also introduce an improvement of the storage and the retrieval algorithm to speed up the retrieval process.

In the retrieval algorithm shown in Fig. 8, all points of P_m is used as a starting point p_0 to examine all cyclic permutations of L_m at the lines 4 and 5. This is because L_m of the retrieval algorithm does not necessarily match L_m of the storage algorithm due to rotations of camera-captured images.

However, the examination of all cyclic permutations can be omitted if the same p_0 is always selected both at the storage and the retrieval processes. We have introduced a selection rule of the starting point to the storage and retrieval algorithm.

The selection rule is shown in Fig. 10. For each point i of m points, an affine invariant $s_{(i)}$ is calculated by combining it with the following three points. If $s_{(j)}$ has the maximum value in $(s_{(0)}\cdots s_{(m-1)})$, the point j is selected as the starting point. In the example of Fig. 10, point 1 is the starting point. If there are two or more equivalent maximum values, the succeeding value $s_{((i+1){\rm mod} m)}$ is used to select one of them. For example, if $s_{(i)} = s_{(j)}$

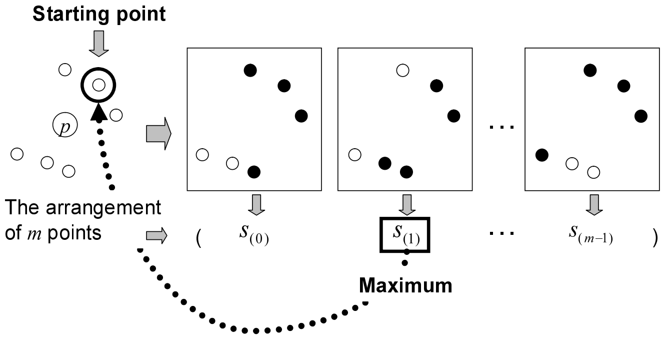
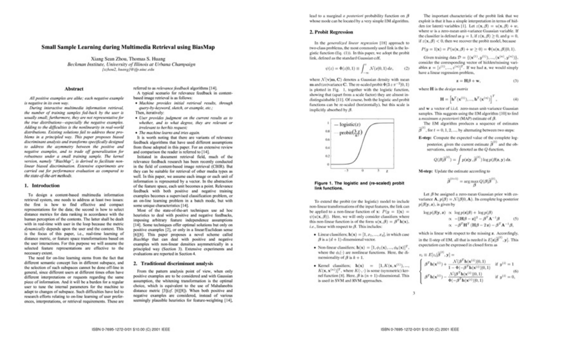

Starting point
p
The arrangement of m points
($s_{(0)}$ $s_{(1)}$ ⋯ $s_{(m-1)}$)
⋯
Maximum

Figure 10. The point which gives maximum $s_{(i)}$ is selected as the starting point.



Small Sample Learning during Multimedia Retrieval using BiasMap

Xiang Sean Zhou, Thomas S. Huang
Beckman Institute, University of Illinois at Urbana Champaign
{xzhou2, huang}@ifp.uiuc.edu

Abstract

All positive examples are alike; each negative example is negative in its own way.

During interactive multimedia information retrieval, the number of training samples fed-back by the user is usually small; furthermore, they are not representative for the true distributions—especially the negative examples. Adding to the difficulties is the nonlinearity in real-world distributions. Existing solutions fail to address these problems in a principled way. This paper proposes biased discriminant analysis and transforms specifically designed to address the asymmetry between the positive and negative examples, and to trade off generalization for robustness under a small training sample. The kernel version, namely "BiasMap", is derived to facilitate non-linear biased discrimination. Extensive experiments are carried out for performance evaluation as compared to the state-of-the-art methods.

1. Introduction

To design a content-based multimedia information retrieval system, one needs to address at least two issues: the first is how to find effective and compact representations for the data; the second is how to select distance metrics for data ranking in accordance with the human perception of the contents. The latter shall be dealt with in real-time with user in the loop because the metric *dynamically* depends upon the user and the context. This is the focus of this paper, i.e., real-time learning of distance metric, or feature space transformations based on the user interactions. For this purpose we will assume the selected feature representations are effective to the necessary extent.

The need for on-line learning stems from the fact that different semantic concept lies in different subspace, and the selection of such subspaces cannot be done off-line in general, since different users at different times often have different interpretations or requests regarding the same piece of information. And it will be a burden for a regular user to tune the internal parameters for the machine to adapt to changes of subspace. Such difficulties have led to research efforts relating to on-line learning of user preferences, interpretations, or retrieval requirements. These are referred to as *relevance feedback* algorithms [14].

A typical scenario for relevance feedback in content-based image retrieval is as follows:

- *Machine provides initial retrieval results, through query-by-keyword, sketch, or example, etc.;*

Then, iteratively:

- *User provides judgment on the current results as to whether, and to what degree, they are relevant or irrelevant to her/his request;*
- *The machine learns and tries again.*

It is worth noting that there are variants of relevance feedback algorithms that have used different assumptions from those adopted in this paper. For an extensive review and comparison the reader is referred to [14].

Initiated in document retrieval field, much of the relevance feedback research has been recently conducted in the field of content-based image retrieval (CBIR). But they can be suitable for retrieval of other media types as well. In this paper, we assume each image or each unit of information is represented by a vector. In the abstraction of the feature space, each unit becomes a point. Relevance feedback with both positive and negative training examples becomes a supervised classification problem, or an on-line learning problem in a batch mode, but with some unique characteristics [14].

Most of the state-of-the-art techniques use ad hoc heuristics to deal with positive and negative feedbacks, imposing arbitrary feature independence assumptions [14]. Some techniques offer optimal solutions but only on positive examples [2], or only in a linear/Euclidean sense [6][8]. This paper proposes a novel scheme called *BiasMap* that can deal with positive and negative examples with non-linear densities asymmetrically in a principled way (Section 3). Extensive experiments and evaluations are reported in Section 4.

2. Traditional discriminant analysis

From the pattern analysis point of view, when only positive examples are to be considered and with Gaussian assumption, the whitening transformation is the optimal choice, which is equivalent to the use of Mahalanobis distance metric [3](*cf.* [6][8]). When both positive and negative examples are considered, instead of various seemingly plausible heuristics for feature-weighting [14],

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE

lead to a marginal *a posteriori* probability function on β whose mode can be located by a very simple EM algorithm.

2. Probit Regression

In the *generalized linear regression* [18] approach to two-class problems, the most commonly used link is the logistic function (Eq. (1)). In this paper, we adopt the probit link, defined as the standard Gaussian cdf,

$$\psi(z) = \Phi(z|0,1) \equiv \int_{-\infty}^{z} \mathcal{N}(x|0,1)\,dx, \qquad (2)$$

where $\mathcal{N}(\mathbf{v}|\mathbf{m},\mathbf{C})$ denotes a Gaussian density with mean $\mathbf{m}$ and (co)variance $\mathbf{C}$. The re-scaled probit $\Phi(3\,z\,\pi^{-2}|0,1)$ is plotted in Fig. 1, together with the logistic function, showing that (apart from a scale factor) they are almost indistinguishable [11]. Of course, both the logistic and probit functions can be re-scaled (horizontally), but this scale is implicitly absorbed by β.

— logistic(z)
— probit($\frac{3}{\pi^2}z$)

Figure 1. The logistic and (re-scaled) probit link functions.

To extend the probit (or the logistic) model to include non-linear transformations of the input features, the link can be applied to a non-linear function of $\mathbf{x}$: $P(y=1|\mathbf{x}) = \psi(u(\mathbf{x},\beta))$. Here, we will only consider classifiers where this non-linear function is of the form $u(\mathbf{x},\beta) = \beta^T\mathbf{h}(\mathbf{x})$, *i.e.*, linear with respect to β. This includes:

- Linear classifiers: $\mathbf{h}(\mathbf{x}) = [1, x_1, ..., x_d]$, in which case β is a $(d+1)$-dimensional vector.
- Non-linear classifiers: $\mathbf{h}(\mathbf{x}) = [1, \phi_1(\mathbf{x}), ..., \phi_k(\mathbf{x})]^T$, where the $\phi_i(\cdot)$ are nonlinear functions. Here, the dimensionality of β is $k+1$.
- Kernel classifiers: $\mathbf{h}(\mathbf{x}) = [1, K(\mathbf{x},\mathbf{x}^{(1)}), ..., K(\mathbf{x},\mathbf{x}^{(n)})]^T$, where $K(\cdot,\cdot)$ is some (symmetric) kernel function [8]. Here, β is $(n+1)$-dimensional. This is used in SVM and RVM approaches.

The important characteristic of the probit link that we exploit is that it has a simple interpretation in terms of hidden (or latent) variables [1]. Let $z(\mathbf{x},\beta) = u(\mathbf{x},\beta) + w$, where w is a zero-mean unit-variance Gaussian variable. If the classifier is defined as $y=1$, if $z(\mathbf{x},\beta) \geq 0$, and $y=0$, if $z(\mathbf{x},\beta) < 0$, then we recover the probit model, because

$$P(y=1|\mathbf{x}) = P(u(\mathbf{x},\beta) + w \geq 0) = \Phi(u(\mathbf{x},\beta)|0,1).$$

Given training data $\mathcal{D} = \{(\mathbf{x}^{(1)},y^{(1)}), ..., (\mathbf{x}^{(n)},y^{(n)})\}$, consider the corresponding vector of hidden/missing variables $\mathbf{z} = [z^{(1)}, ..., z^{(n)}]^T$. If we had $\mathbf{z}$, we would simply have a linear regression problem,

$$\mathbf{z} = \mathbf{H}\beta + \mathbf{w}, \qquad (3)$$

where $\mathbf{H}$ is the *design matrix*

$$\mathbf{H} = \left[\mathbf{h}^T(\mathbf{x}^{(1)}), ..., \mathbf{h}^T(\mathbf{x}^{(n)})\right]^T, \qquad (4)$$

and $\mathbf{w}$ a vector of i.i.d. zero-mean unit-variance Gaussian samples. This suggests using the EM algorithm [10] to find a *maximum a posteriori* (MAP) estimate of β.

The EM algorithm produces a sequence of estimates $\widehat{\beta}^{(t)}$, for $t = 0, 1, 2, ...$, by alternating between two steps:

E-step: Compute the expected value of the complete log-posterior, given the current estimate $\widehat{\beta}^{(t)}$ and the observations, usually denoted as the Q-function,

$$Q(\beta|\widehat{\beta}^{(t)}) = \int p(\mathbf{z}|\mathbf{y},\widehat{\beta}^{(t)}) \log p(\beta|\mathbf{z},\mathbf{y})\,d\mathbf{z}.$$

M-step: Update the estimate according to

$$\widehat{\beta}^{(t+1)} = \arg\max_{\beta} Q(\beta|\widehat{\beta}^{(t)}).$$

Let β be assigned a zero-mean Gaussian prior with covariance $\mathbf{A}$, $p(\beta) = \mathcal{N}(\beta|0,\mathbf{A})$. Its complete log-posterior $p(\beta|\mathbf{y},\mathbf{z})$, is given by

$$\begin{aligned} \log p(\beta|\mathbf{y},\mathbf{z}) &\propto \log p(\mathbf{z}|\beta) + \log p(\beta) \\ &\propto -\|\mathbf{H}\beta - \mathbf{z}\|^2 - \beta^T\mathbf{A}^{-1}\beta \\ &\propto -\beta^T\mathbf{H}^T(\mathbf{H}\beta - 2\mathbf{z}) - \beta^T\mathbf{A}^{-1}\beta, \end{aligned} \qquad (5)$$

which is linear with respect to the missing $\mathbf{z}$. Accordingly, in the E-step of EM, all that is needed is $E[\mathbf{z}|\widehat{\beta}^{(t)},\mathbf{y}]$. This expectation can be expressed in closed form as

$$v_i \equiv E[z_i|\widehat{\beta}^{(t)},\mathbf{y}] = \begin{cases} \beta^T\mathbf{h}(\mathbf{x}^{(i)}) + \dfrac{\mathcal{N}(\beta^T\mathbf{h}(\mathbf{x}^{(i)})|0,1)}{1-\Phi(-\beta^T\mathbf{h}(\mathbf{x}^{(i)})|0,1)} & \text{if } y^{(i)} = 1 \\ \beta^T\mathbf{h}(\mathbf{x}^{(i)}) - \dfrac{\mathcal{N}(\beta^T\mathbf{h}(\mathbf{x}^{(i)})|0,1)}{\Phi(-\beta^T\mathbf{h}(\mathbf{x}^{(i)})|0,1)} & \text{if } y^{(i)} = 0, \end{cases} \qquad (6)$$

3

ISBN 0-7695-1272-0/01 $10.00 (C) 2001 IEEE

Figure 11. Examples of images in the database.

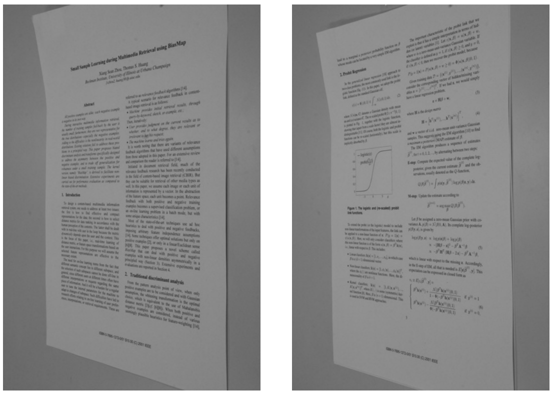


Figure 12. Examples of query images.

and both are the maximum, then the value of $s_{((i+1)\bmod m)}$ and $s_{((j+1)\bmod m)}$ are compared. If $s_{((i+1)\bmod m)}$ is larger, the point i is selected as the starting point. In the case that $s_{((i+1)\bmod m)} = s_{((j+1)\bmod m)}$ the succeeding values $s_{((i+2)\bmod m)}$ and $s_{((j+2)\bmod m)}$ are likewise examined.

5. Experimental results

In order to examine effectiveness of improvements introduced in this paper, we investigated performances of the original and improved versions of LLAH. To clarify the effect of memory reduction stated in Sect. 3. and that of speeding up stated in Sect. 4., we measured the required amount of memory, processing time and accuracy of three versions of LLAH: the first one is the original LLAH, the second one is the memory reduced, and the third one is the memory reduced and speeded up version. The third version is the proposed method in this paper.

Document images stored in the database were images converted with 200 dpi from PDF files of single- and double-column English papers collected mainly from CD-ROM proceedings. Examples of images in the database are shown in Fig. 11. Query images were captured using a digital camera with 6.3 million pixels. As shown in Fig. 12, they were captured from a skew angle (about 45 degree). Since the angle with which query images are captured (45 degree) is different from that of the images in the database (90 degree), experiments performed with these query images and the database would demonstrate robustness of the proposed method to perspective distortion. Note that the query images suffer from severer distortion than those of [2]. Experiments were performed on a workstation with AMD Opteron 2.8GHz CPUs and 16GB memory. Parameters[1] were set to $n = 7, m = 6, k = 15$. H_{size} was set to 1.28×10^8.

5.1. Required amount of memory

Figure 13 shows the amount of required memory of the three version of LLAH with 100, 1,000 and 10,000 pages databases. The original version of LLAH required 5 times larger amount of memory than improved ones. Moreover,

[1] Some experimental results with different n and m are found in [2].

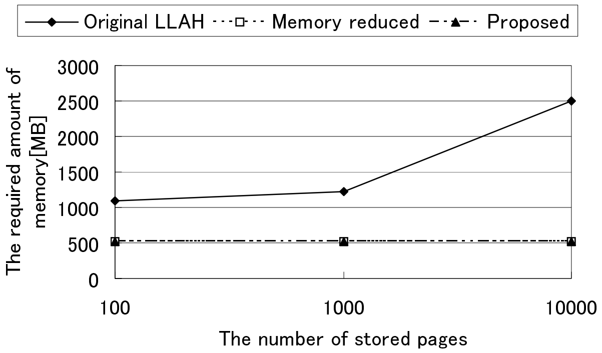


Figure 13. The relationship between the number of stored pages and the required amount of memory.

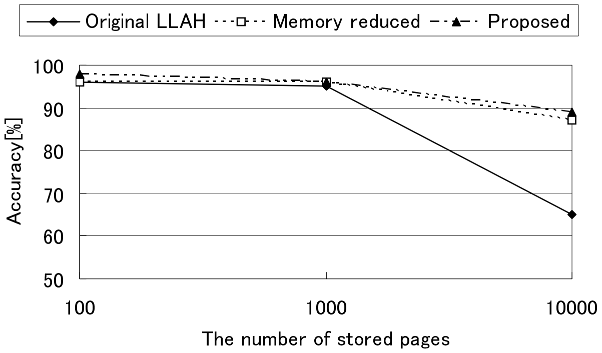


Figure 15. The relationship between the number of stored pages and accuracy.

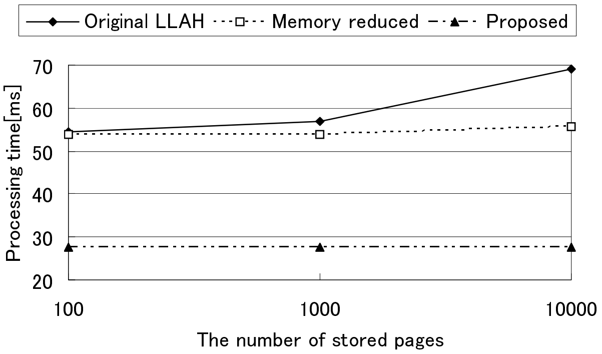


Figure 14. The relationship between the number of stored pages and processing time of retrieval.

the amount increased with increasing the number of stored pages. Since the original LLAH adopts the linked list as the form of the database, more stored pages result in more amount of memory. On the other hand, in the memory reduced versions, the required amount of memory was constant regardless of the number of stored pages. Since these versions adopt a simple hash table as the form of the database, required memory is that for a hash table of a fixed size.

5.2. Processing time of retrieval

Figure 14 shows processing time by each version of LLAH. The proposed version of LLAH realizes reduction of the processing time by about 60%. This is because the number of computed invariants and access to the database have been reduced by skipping the calculation of all possible cyclic permutations.

It is also confirmed that processing time of the memory reduced version was almost constant regardless of the number of stored pages as contrasted to the original version. This is because access to a simple hash table requires constant computations while access to linked lists requires computations in proportion to the number of stored items.

5.3. Accuracy

Figure 15 shows accuracy of retrieval of each version of LLAH. Improved versions demonstrated higher performance in terms of accuracy in addition to time and memory efficiency. In contrast to the original LLAH which showed less accuracy with a larger number of stored pages, improved versions showed higher accuracy although they also had a decrease in accuracy with 10,000 stored pages.

Although the improvements proposed in this paper were not intended to improve accuracy, they realized higher accuracy. We consider the reason is that erroneous votes are decreased as a result of removal of less important features. In the improved versions, features which cause collisions are removed from the database. Since such features tend to cause erroneous votes, removal of them results in higher accuracy.

6. Related work

In LLAH, document images are retrieved based on local arrangements of feature points. Therefore it can be classified into an image retrieval method using local features. There have been various types of image retrieval methods using local features. They can be classified into two types: one based on complex features such as SIFT and the other based on simple features such as feature points.

Video Google [6] is one of the image retrieval methods using complex local features. In Video Google, a codebook is created by clustering SIFT features extracted from images prior to retrieval. Retrieval is performed based on vector quantized SIFT features of query images using the codebook. In order to realize accurate retrieval, a large codebook is needed. However use of a large codebook results in long processing time since nearest neighbor search takes much time. It is also a problem that calculation of SIFT features needs much computation.

Geometric Hashing(GH) [7] is well known as an image retrieval method based only on feature points. In GH, features are calculated by combining feature points in order to realize stability of features. For example, O(N^4) computation is required for the retrieval under affine distortion where N is the number of feature points. Therefore the number of combinations becomes enormous when images have many feature points. Since document images have many feature points, it is prohibitive to apply GH to retrieval of document images. For more details, see [8].

7. Conclusion

In this paper, we have introduced improvements to the LLAH. The required amount of memory was decreased by removal of unnecessary features and simplification of structure of the hash table. Processing time of retrieval was shortened by the improvement of the retrieval algorithm. From the experimental results, we have confirmed reduction of the required amount of memory by 80% and shortening of the processing time of retrieval by 60%. It is also confirmed that the improvements bring higher accuracy. From these results, we can conclude the proposed improvements realize better scalability and extensibility to applications which require hi-speed retrieval.

Our future tasks include improvements of feature point extraction process. The feature point extraction process can become a bottleneck of the whole image retrieval process using LLAH since it requires heavy image processing. Since LLAH currently covers only English document images, it is also necessary to add other objects in the target of LLAH.

Acknowledgement

This research is supported in part by Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science (19300062, 19·7621).

References

[1] D. Doermann, “The indexing and retrieval of document images: a survey”, Computer Vision and Image Understanding, vol. 70, no. 3, pp.287–298, 1998.

[2] T. Nakai, K. Kise, and M. Iwamura, “Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval”, Lecture Notes in Computer Science (7th International Workshop DAS2006), vol. 3872, pp.541–552, 2006.

[3] J. Liang, D. Doermann, and H. Li, “Camera-based analysis of text and documents: a survey”, IJDAR, vol. 7, pp.84–104, 2005.

[4] P. Clark, and M. Mirmehdi, “Recognising text in real scenes”, IJDAR, vol. 4, pp. 243–257, 2002.

[5] C. H. Lampert, T. Braun, A. Ulges, D. Keysers, and T. M. Breuel, “Oblivious document capture and real-time retrieval”, Proc. CBDAR2005, pp.79–86, 2005.

[6] J. Sivic and A. Zisserman, “Video google: a text retrieval approach to object matching in videos”, Proc. ICCV2003, vol. 2, pp.1470–1477, 2003.

[7] H. J. Wolfson and I. Rigoutsos, “Geometric hashing: an overview”, IEEE Computational Science & Engineering, vol. 4, no. 4, pp.10–21, 1997.

[8] M. Iwamura, T. Nakai and K. Kise, “Improvement of retrieval speed and required amount of memory for geometric hashing by combining local invariants”, Proc. BMVC2007, 2007 [to appear].

Mobile Retriever - Finding Document with a Snapshot

Xu Liu
Institute for Advanced Computer Studies
University of Maryland
liuxu@cs.umd.edu

David Doermann
Institute for Advanced Computer Studies
University of Maryland
doermann@umd.edu

Abstract

In this paper we describe a camera based document image retrieval system which is targeted toward camera phones. Our goal is to enable the device to identify which of a known set of documents it is "looking at". This paper provides two key contributions 1) a local context descriptor that effectively rules out irrelevant documents using only a small patch of the document image and 2) a layout verification approach that boosts the accuracy of retrieval even under severe degradation such as warping or crinkling. We have implemented the mobile retriever client on an iMate Jamin camera phone and tested it with a document database of 12742 pages. Experiments show that our verification approach clearly separates successful retrievals from unsuccessful retrievals.

1. Introduction

1.1. Motivation

The research in this paper is motivated by two facts. First, the trends toward a paperless office is leading to large quantities of documents existing in both electronic form (being born digital or being digitalized) and in hard copy (newspaper, magazine, etc.). Most of documents have digital version. Second, the number of camera phone users has increased tremendously in recent years. According to Garner's report[4] 48% of cellular phones had cameras in 2006, and is projected to increase to 81% by 2010. The camera phone is an ideal platform for content based retrieval systems [5] since it is easy to carry, it has the computational power of image processing and is linked to the wireless network. In this paper, we provide a way to enable camera phones (or other camera enabled devices) to serve as the input device for visually querying a document image database. More specifically, our document retrieval is based on a partial snapshot (Fig. 1 (b)) of the page from an unconstrained viewing angle (Fig. 1 (a)) , with the goal of finding the original page (Fig. 1 (c)) in the database. Finding the page will enable many interesting applications for mobile access.

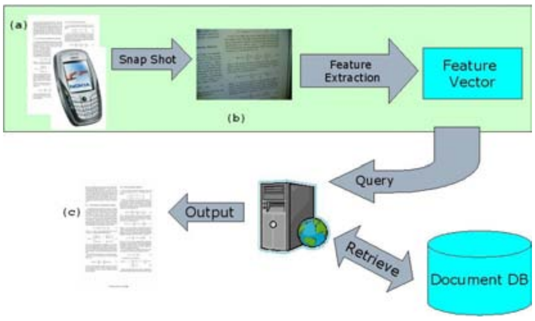


Figure 1. System

1.2. Use Scenario

Researchers often store a large number of digital documents on their computers. Often he has one printed paper and hopes to find the original PDF file, the simplest way is to identify the title of the paper, type it into a local/web search engine and look for the correct record from the results. On the desktop or laptop this approach is reasonable but for mobile devices it is cumbersome. Our approach simplifies this procedure into one snapshot. The original electronic document can be retrieved immediately when the camera phone "sees" the paper, even if the image quality is not good enough to "read" it.

Publishers will know what their reader reads if the reader is willing to take a snapshot of the article. The readers can comment, annotate and send feedback immediately.

The visually impaired can listen to the audio version of the article if its snapshot can be retrieved. The assumption is that the document is pre-stored in our database and the audio version is ready or can be synthesized using text to speech.

Watermarking usually requires special modification to document texts and can not be stably read by a camera. Our approach can be used as a natural watermark for every document. It is unique and can be read by a camera phone.

1.3. Related work

Various approaches has been explored for image based document retrieval. In [6], Hull proposed a series of distortion-invariant descriptors allowing robust retrieval against re-formatting, re-imaging and geometric distortion. In [1], Cullen et al. use texture cues to retrieve documents from a database. In [8], Tan et al. measure document similarity by matching partial word images. In [7] Kameshiro et al. describe the use of an outline of the character shape, that tolerates recognition and segmentation errors for document image retrieval. Our approach is most closely related to the system proposed by Nakai and Kise et al. in [9] and [10]. In their approach, combinations of local invariants are hashed into a large hash table. Retrieval is accomplished by voting from this hash table and they are able to obtain an accuracy of 98% over 10000 documents. However the combinations of local invariants result in a very large feature vector. Furthermore, the query image must cover a large portion of the page which is sometimes hard to enforce especially with a camera phone. Camera phones are usually equipped with lower-end CMOS cameras; when capturing the whole page, the resolution might be too low to have the words separated. In our approach we have loosened the requirements of capturing, requiring only about 1/8 of a page. This makes the problem harder because the captured image could be from anywhere of the page, and a stable distribution of features cannot be expected because a large portion of the page may be absent.

Since we want our retrieval to be robust against perspective distortion, occlusion, uneven lighting, and even crinkled pages, we cannot use global configurations of feature points which might be partially missing or changed by degradations. Like Kise [9], we use local features which we call the “layout context”. The rest part of this paper is organized as follows. A brief description of how we process the camera phone captured image in Section 2 is followed by a detailed explanation of layout context is contained in Section 3. After gathering a large number of layout contexts we cluster them to build a lexicon (Section 4) to index and re-rank the pages in database. A global layout verification step is introduced in Section 5, followed by experiments in Section 6. We discuss the shortcomings and future work in Section 7.

2. Image Processing

Before feature extraction, we need to separate the foreground contents from the background of the page, i.e. we binarize the image to identify the text. Camera phone images may be captured from various angles, and under various lighting conditions. Therefore, a global binarization will typically not work. We employ Niblack’s[11] adaptive binarization method and then extract connected components from the image. A typical binarized image is shown in Fig. 2 and is sufficient for identifying words and components.

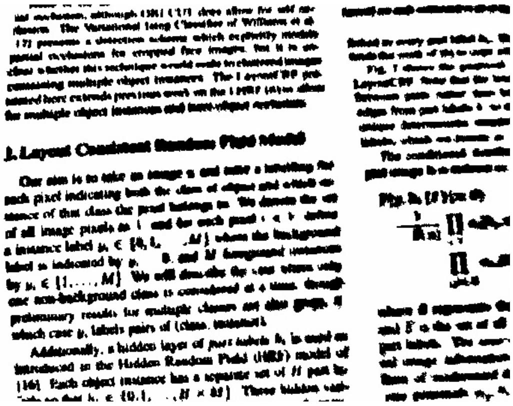


Figure 2. Adaptive binarized snapshot

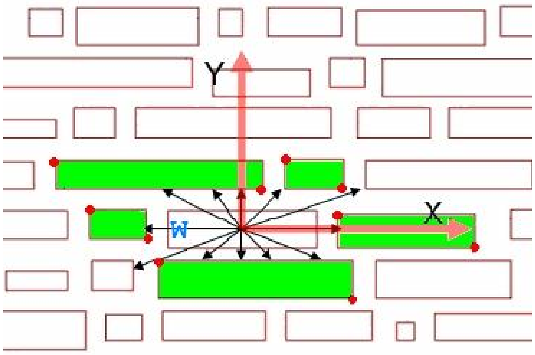


Figure 3. An Example of Layout Context

Although it will be very difficult, if not impossible, to extract stable and distinctive features from a single connected component (word), the relationships between components speak their own language - a language of contexts. Our goal is to extract the lexicon of this language,and index and retrieval documents using this lexicon. The first task is to define a “word” in this lexicon, i.e. the layout context.

3. Layout Context

We begin with an ideal image with no perspective distortion. In Fig. 3 each rectangle shows a bounding box of a word. To extract the layout context of a word w in Fig. 3, suppose we begin at the center of the word and look for the most visible n neighbors. Fig. 3 shows, for $n = 5$, using 5 green rectangles. The visibility is defined by the angle of the view and the top n can be extracted using an efficient computational geometry algorithm with complexity linearly bounded by the total number of nearest m neighbors (it is safe to choose $m = 10n$). The top n visible neighbors are invariant to rotation and the percentage of view angles that a neighbor word occupies will not be effected by rotation. We put the coordinate system origin at the center of w with the X-axis parallel to the baseline of w and define the unit metric using the width of w. Under this coordinate system, the coordinates of the n most visible neighbors are invariant to similarity transformations.

- Translation: the original point always stays at the center of word w.
- Rotation: the X-axis always falls along the direction of the text lines.
- Scale: distortions are normalized by the width of word w. To use width of w as a stable factor of normalization,w must have an aspect ratio greater than a threshold (3 for example). This condition is satisfied by a large portion of words that have more than 3 characters.

With $n = 5$, a layout context is a vector that occupies $5 \times 2 \times 2 = 20$ bytes data (5 context words, 2 corners per word, 2 bytes per corner). When a document image is captured using a camera phone, it undergoes perspective transform, but locally can still be approximated by a similarity transform. For a similarity transform, scale, translation and rotation have to be normalized. We detect the baseline of text lines by finding the lower boundary of every connected component and the locally rotate text lines into the horizontal direction. After this, the scaling normalization is the same as for a perfect image.

4. Building the lexicon

As stated above, these low resolution documents speak a language of contexts. To understand this language, we must first build its lexicon, i.e. the dictionary of "visual words" [12] . We define the lexicon to be a set of representative layout contexts extracted from a training set of layout contexts. For example, we used 2000 pages randomly selected from the proceedings of CVPR04, 05, 06. From these 2000 pages we collect approximately 600 layout contexts from each page, for a total of 120548 layout contexts. For two reasons we cannot directly use these 120548 layout contexts as the lexicon. First, such a large lexicon will make the indexing procedure slow since we need to index each layout context by its nearest neighbors. Second, such a lexicon has layout contexts which are very similar; the nearest neighbor search could result in misclassification. In order to reduce the dimension, we run a mean-shift clustering on the layout contexts that results in a lexicon of containing 10042 clusters <10% of the original size.

5. Verification

The layout contexts extracted from the camera captured image may not be exactly the same as the one stored in the database for the following reasons:

- The word segmentation can be erroneous because of uneven lighting or inaccurate binarization, neighbor words could be touching and long words could be segmented. Segmentation inconsistency also occurs in the presence of non-text elements such as formulas and figures.
- On the boarder area of the camera captured image, the layout context may be different from the layout context stored in the database because some neighbor words are missing in the camera captured image.
- The document page might not be as perfectly flat as its digital version, warping and crinkling might destroy the planar invariants.

After sorting the documents by the coverage of layout contexts, therefore, a verification step is required and this step is our key contribution. To verify point set matches, RANSAC[3] is a classical model based algorithm. But RANSAC suffers most from non-rigid degradation since it is based on a model of transform, i.e. for plane-to-plane matching, this model is a projective transform (or homography). The assumption is that, all the inliers of matches must fit in this model exactly. But when a paper is warped or crinkled, which is very common, the model collapses since a homography can no longer be used to map from one point set to another. To allow a non-rigid transform, a more elastic method such as soft assign[2] might be used. However, soft assign is an iterative method which could take a long time to converge.

We propose a triplet based point matching algorithm which is robust against projective transforms, deformations and occlusions. Consider three points (A, B, C) on a 2D planar surface with homogeneous coordinates $(X_A, Y_A, 1)$, $(X_B, Y_B, 1)$ and $(X_C, Y_C, 1)$, their orientation is defined as

$$Sign(\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}) \qquad (1)$$

where

$$Sign(X) = \begin{cases} 1 \cdots X \geq 0 \\ -1 \cdots X < 0 \end{cases}$$

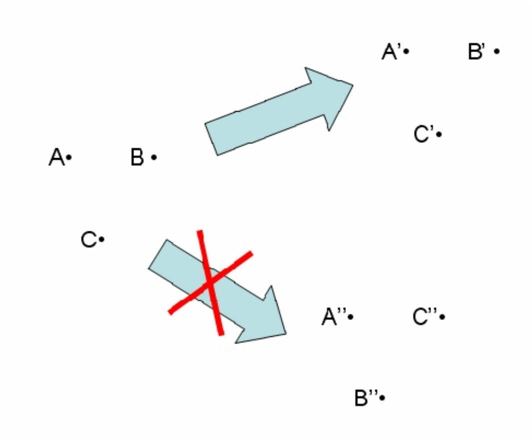


Figure 4. Possible Triplet Matches

When this surface is bent or viewed from another view angle, these three points appears as A', B' and C' and we have

$$Sign(\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}) \times Sign(\begin{vmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{vmatrix}) = 1 \tag{2}$$

which means the orientation of (A, B, C) is consistent with (A', B', C'). On the contrary, (A, B, C) is inconsistent with (A', B', C') when

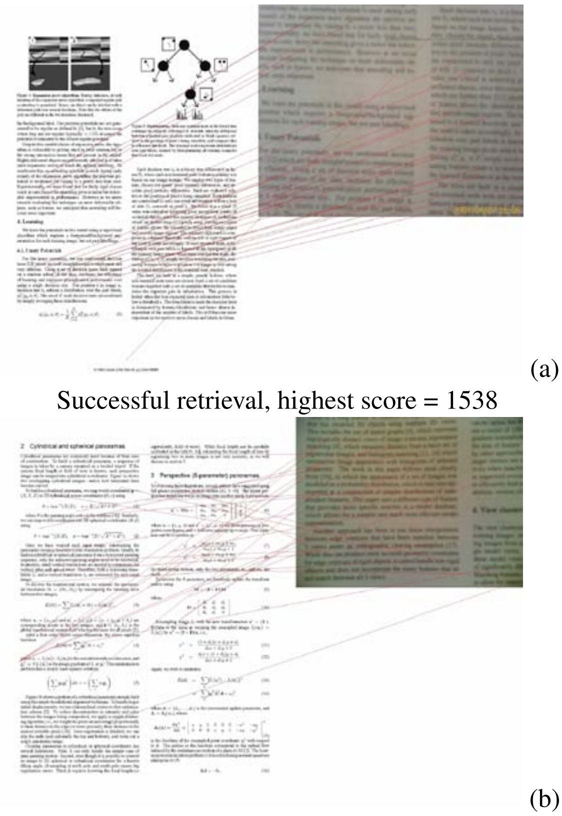
$$Sign(\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}) \times Sign(\begin{vmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{vmatrix}) = -1 \tag{3}$$

When a point set S is matched to another point S', we define the score of this match as

$$\sum_{A,B,C \in S} (Sign(\begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix}) \times Sign(\begin{vmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{vmatrix})) \tag{4}$$

An ideal match from a one point set to another n-point set has a score of $\binom{n}{3}$ when every triplet is consistent with its match. The worst match score is $-\binom{n}{3}$ (mirrored).

In order to obtain the best match between two sets, a maximum flow or Hungarian algorithm can be used, but such an algorithm has a complexity of $O(v^3)$, where v is the number of vertices of the bi-partition graph (often greater than 600 for a single page). Since we will apply this verification step to a list of candidate pages, it consumes most of the runtime and must be efficient. We use a greedy algorithm to find an approximate match instead. Consider the two point sets as a bipartite graph and the value of each edge is the Euclidian distance between the layout contexts of its two vertices. We find the edge with the smallest value, match its two vertices, remove this edge together with its vertices, and repeat this procedure m times to find m pairs of point matches. The score of these m matches is between $-\binom{m}{3}$ and $\binom{m}{3}$.



(a)

Successful retrieval, highest score = 1538

(b)

Failed to retrieve, highest score = 332

Figure 5. Example matches with point correspondences, m=30

6. Implementation and Experiment

We have collected the proceedings of CVPR04, 05, 06 and ICCV05, 12742 pages in total. Table 1 shows the number of pages from each proceedings. From every page we extract layout contexts and each layout context is indexed by its nearest neighbor from the lexicon. Every page is complied into a bag of indexes with their coordinates. The coordinates are for the verification step.

When retrieving, the camera captured image is also complied into a bag of indexes with coordinates after rotation normalization. The pages in the document database are sorted by their coverage of the bag of query indexes. Notice that we are only interested in the top part of this sorted list of pages, most of the pages are eliminated immediately. We verify the top 1000 pages from this sorted list and the page that gets the highest score is the result of our retrieval. Fig. 5 shows a successfully retrieval (a) and an unsuccessful (page not in databse) retrieval (b). From the point-to-point correspondence we can see that the successful retrieval has almost all lines "parallel" i.e. they intersect at a vanish point and has a high score (a), while the unsuccessful matches point in arbitrary directions and have a low score (b).

Our mobile retriever is implemented on an iMate Jamin (Windows Mobile 5.0) phone using a camera with 1024×1280 resolution. From each captured image we extract 100 layout contexts, each of which takes about 24 bytes together with its coordinates, and in total approximately 2.4KB is required per query. For simplification and logging purpose, in our current implementation we upload the image to the server and extract features on the server side. In the future, the image capturing, processing and feature extraction (green box in Fig. 1) can all be done on the mobile device and the communication from device to server will take less than one second via GPRS/EDGE/CDMA network.

To test the performance of our system, we randomly select 50 pages from the database and 50 pages that are not in the database and capture pictures of these pages as queries for retrieval. Among the first 50 captures, 45 were successfully retrieved; among the second 50, as expected, all are rejected. We show a scatter plot of successful and unsuccessful retrieves in Fig. 6 with their scores and ranks. We can see a clear gap between successful retrieval and rejection. Therefore when a page has a score greater than 800, we have a high confidence that it is a correct retrieval. When a page has a score less than 400 it must not be a correct retrieval. By setting a proper threshold, we can achieve an 100% accurate retrieval. However, a page with high score may not have a high rank since some of the layout contexts can be mismatched and this mismatch will only be corrected during verification step. Fig. 7 shows two successful retrievals and two failed retrievals. Our approach is robust under crinkled and warped degradations. Since it relies on the layout of words, it fails when there is a small amount of text present in the captured image. We also approximate projective transform locally using similarity transform, so it may fail when perspective distortion is too strong.

7. Conclusion and future work

In this paper we present an end-to-end system that retrieves original documents from a camera phone captured sample. We use a distinctive local "layout context" descriptor to represent features of the document image and we verify the retrievals using triplets orientation which is robust to page or imaging distortion. This verification draws a clear

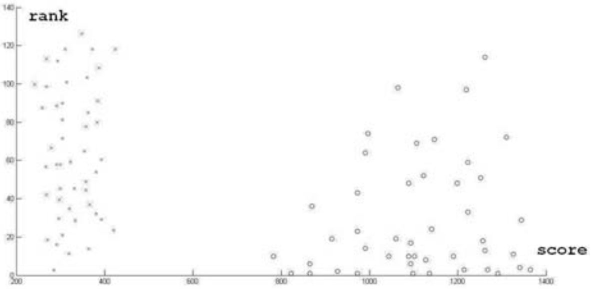
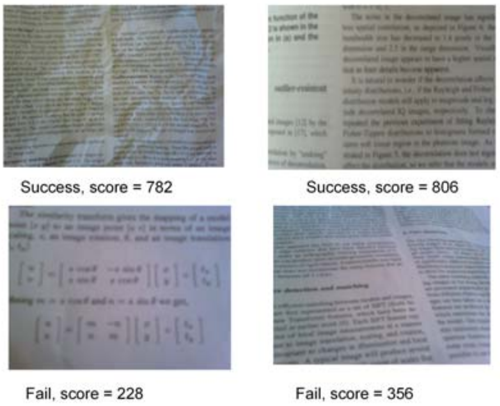


Figure 6. Score and rank of retrieval, dot: success, cross: rejection


Success, score = 782

Success, score = 806

Fail, score = 228

Fail, score = 356

Figure 7. Successful and unsuccessful retrievals

Table 1. Data Collection

CVPR04	3397 pages
CVPR05	3501 pages
CVPR06	4001 pages
ICCV05	1843 pages
TOTAL	12742 pages

gap between successful and unsuccessful retrievals. A draw back of this verification is that it has to be applied to every page candidate and takes most of the runtime. On a Pentium 4, 2GHz CPU, a retrieval might take up to 20 seconds in going through 200 candidates. In future work, this verification may be replaced by a hashing of triplets which can accelerate speed. Another limitation with our approach and with most of the existing approaches is that, they are based on word and therefore targeted for Latin languages. For Asian languages such as Chinese, Japanese and Korean a new local feature descriptor has to be designed but the verification can still be applied.

References

[1] J. Cullen, J. Hull, and P. Hart. Document image database retrieval and browsing using texture analysis. *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 718–721, 1997.

[2] P. David, D. DeMenthon, R. Duraiswami, and H. Samet. SoftPOSIT: Simultaneous Pose and Correspondence Determination. *International Journal of Computer Vision*, 59(3):259–284, 2004.

[3] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[4] Gartner. Nearly 50 percent of worldwide mobile phones will have a camera in 2006 and 81 percent by 2010. *Gartner*.

[5] J. Hare and P. Lewis. Content-based image retrieval using a mobile device as a novel interface. *Storage and Retrieval Methods and Applications for Multimedia 2005. Proceedings of the SPIE,*, 5682:64–75, 2004.

[6] J. Hull. Document image matching and retrieval with multiple distortion-invariant descriptors. *Document Analysis Systems*, pages 379–396, 1995.

[7] T. Kameshiro, T. Hirano, Y. Okada, and F. Yoda. A document image retrieval method tolerating recognition andsegmentation errors of OCR using shape-feature and multiple candidates. *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*, pages 681–684, 1999.

[8] Y. Lu and C. L. Tan. Information retrieval in document image databases. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1398–1410, 2004.

[9] T. Nakai, K. Kise, and M. Iwamura. Hashing with Local Combinations of Feature Points and Its Application to Camera-Based Document Image Retrieval. *Proc. CBDAR05*, pages 87–94, 2005.

[10] T. Nakai, K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. *DAS06*, pages 541–552, 2006.

[11] W. Niblack. *An introduction to digital image processing*. Strandberg Publishing Company, Birkeroed, Denmark, Denmark, 1985.

[12] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477, 2003.

Section III

Dewarping

Perspective rectification for mobile phone camera-based documents using a hybrid approach to vanishing point detection

Xu-Cheng Yin Jun Sun Satoshi Naoi
Fujitsu R&D Center Co. Ltd, Beijing, China
{xuchengyin; sunjun; naoi}@cn.fujitsu.com

Yusaku Fujii Katsuhito Fujimoto
Fujitsu Laboratories Ltd, Kawasaki, Japan
{fujii.yusaku; fujimoto.kat}@jp.fujitsu.com

Abstract

Documents captured by a mobile phone camera often have perspective distortions. In this paper, a fast and robust method for rectifying such perspective documents is presented. Previous methods for perspective document correction are usually based on vanishing point detection. However, most of these methods are either slow or unstable. Our proposed method is fast and robust with the following features: (1) robust detection of text baselines and character tilt orientations by heuristic rules and statistical analysis; (2) quick detection of vanishing point candidates by clustering and voting; and (3) precise and efficient detection of the final vanishing points using a hybrid approach, which combines the results from clustering and projection analysis. Our method is evaluated with more than 400 images including paper documents, signboards and posters. The image acceptance rate is more than 98% with an average speed of about 100ms.

1. Introduction

Recently, camera-based document analysis becomes a hot research field [6][10]. With widespread usage of the cheap digital ***cam***era built-in the ***mobile*** phone (***MobileCam*** in abbreviation thereafter) in people's daily life, the demand for simple, instantaneous capture of document images emerges. Different from the traditional scanned image, lots of the MobileCam-based document images have perspective distortions (see Fig. 6(a)(b)). Consequently, rectifying MobileCam-based perspective document images becomes an important issue.

As a general computer vision problem, most perspective correction methods rely on vanishing point detection. And these methods involve extracting multiple lines and their intersections, or using texture and frequency knowledge [4][11]. In document analysis, there are also various works on correction of perspective documents captured by general digital cameras [2][3][7][8][9][12][13]. Many of these methods use document boundaries and text lines to vote and detect vanishing points. And other methods take advantage of information of text lines and character tilt orientations.

We divided the methods of vanishing point detection into two sub groups: direct approaches and indirect approaches. The direct approaches directly analyze and calculate on image pixels, such as projection analysis from a perspective view for horizontal vanishing point detection [2]. These approaches have rather good precisions. But the search space in such approaches is an infinite 2D space, and a full or partial search of the space is computationally expensive, even impossible. The indirect approaches convert the original space into a clue space, and search the vanishing point in that new small space. Most of the indirect approaches involve extracting multiple straight or illusory lines[1] and voting vanishing points by model fitting. To calculate the horizontal vanishing point, some methods explicitly fit a line bundle in the linear clue feature space [8][9]. Some researchers use the spacing between adjacent horizontal lines of text to vote the vertical vanishing point [2][8]. Lu et al. use character stroke boundaries and tip points to correct perspective distortions based on multiple fuzzy sets and morphological operators [7]. Liang et al. use the equal text line spacing property to calculate and vote vanishing points, and they suggest their method can be applied on mobile devices [12][13]. These indirect approaches are time efficient. However, the model fitting in these methods are sensitive.

Moreover, in MobileCam-based document analysis, the captured images are always in low resolution with blurring, and the captured text contains often a partial portion of the whole document.

In conclusion, there are two main challenges for rectifying MobileCam-based perspective documents. First, the rectifying engine should be highly precise with fast

[1] The straight and illusory lines used in this paper are similar to the hard and illusory lines respectively termed in [9].

speed. The above methods can't cover the two issues well at the same time. Second, a MobileCam-based image is usually a partial portion of a whole document with few document boundaries. But many traditional methods mainly rely on document boundaries for correction.

Therefore, the traditional direct and indirect approaches have limitations for practical situations. To solve the above problems aiming at a practical MobileCam application, we focus on a fast and robust method for rectifying general perspective documents. First, we propose a hybrid approach for robust real-time vanishing point detection by integrating the direct and indirect approaches efficiently. As for the second challenge, we utilize horizontal text baselines and character tilt orientations as illusory lines to vote and compute vanishing points. Since the character strokes are used in line detection, paper boundaries are not necessary.

Our rectifying method is described in Fig. 1, where preprocessing includes grayscale image conversion, thresholding, and edge calculation, et al. The straight lines, horizontal text baselines, and character vertical strokes all are extracted by connected component analysis, heuristic rules and statistical analysis. Then, the hybrid approach clusters and detects the horizontal and vertical vanishing points respectively.

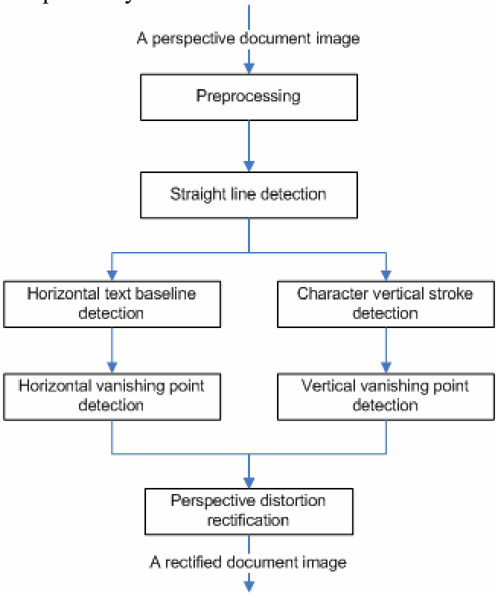

A perspective document image
Preprocessing
Straight line detection
Horizontal text baseline detection
Character vertical stroke detection
Horizontal vanishing point detection
Vertical vanishing point detection
Perspective distortion rectification
A rectified document image

Fig. 1. The flowchart of our method for perspective rectification.

The main contributions of this paper are as follows. The first contribution is a hybrid approach to vanishing point detection, which combines a greedy "indirect" method with a high-quality pixel-based "direct" method. The fast indirect method gives a set of candidates, which are refined afterwards using a direct approach. And the decision is made by linearly combining these two methods. The second contribution is the clustering learning for vanishing point candidates in the indirect method, which fast selects potential vanishing point candidates in the intersection point distribution from all straight and illusory lines. The third contribution is a robust and fast perspective distortion image rectifier, which is a working system for applications of mobile phone camera-based document images.

The remainder of this paper is organized as follows. Section 2 introduces the basic principle of our method. Section 3 explains how to extract the lines and the strokes in detail. In Section 4, we show the strategy of vanishing point detection. Section 5 is the experiments and result analysis. Finally we conclude the paper in Section 6.

2. Basic principle for vanishing point detection

The vanishing point (horizontal or vertical) in a 2D space can be described as (v_x, v_y), where v_x and v_y are the X and Y coordinates respectively in the 2D Euclidean space,

$$\Re^2 = \{(x,y) \mid -\infty < x < +\infty, -\infty < y < +\infty\}, \tag{1}$$

where (0,0) is the center point of the image. In general, the vanishing point does not lie within the image itself.

Generally speaking, vanishing point detection is to find a proper point according to an optimization process in the image plane. That is to say,

$$(v_x, v_y) = \arg\max_{(x,y)\in\Re^2} f(x,y),$$

where $f(x,y)$ is the profit function for the optimization.

For the direct approaches for vanishing point detection, the search space is $\Re^2$ (equation (1)). Obviously, search in such a space is computationally expensive.

In this paper, we propose a novel and hybrid approach for vanishing point detection. Our approach first votes and clusters line intersections into vanishing point candidates. This process belongs to an indirect approach. Then projection analysis from perspective views on these candidates is performed, which is a direct approach. The vanishing point is obtained by an optimization of a function based on the previous two steps. The function can be expressed as following:

$$g(x,y) = G(f_{indirect}(x,y), f_{direct}(x,y)), \tag{2}$$

where $f_{indirect}(x,y)$ and $f_{direct}(x,y)$ are the profit functions for the indirect and direct approaches respectively.

For vanishing point detection, first, we locate all straight and illusory lines. Then calculate all intersections for every line pair. The set of the resulting intersection points is

$$Set(Pt) = \{(x_1,y_1),(x_2,y_2),\ldots,(x_{N_{PT}},y_{N_{PT}})\},$$

where N_{PT} is the number of intersections. These points are partitioned into several groups by a clustering algorithm. Therefore, we get a new space $\boldsymbol{S}$,

$$S = \{S_i \mid S_i \subset \Re^2, i = 1,\ldots, N\}.$$

The center point of each cluster is regarded as a typical representation of its sub region, which is

$$C = \{c_i \mid c_i \in S_i, i = 1,\ldots, N\}. \tag{3}$$

Rather than searching on the whole space in $\Re^2$, we search on the representative point set in $\boldsymbol{C}$ for speed up. Since the point set in $\boldsymbol{C}$ is representative enough, the searched maximum in $\boldsymbol{C}$ is a good approximation to the global maximum.

Consequently, the final resulting vanishing point is given by

$$(v_x, v_y) = \operatorname*{argmax}_{(x,y)\in C} g(x,y).$$

where C is defined in Equation (3).

Now, we perform a direct approach (e.g., projection analysis from a perspective view) on the new search space. Compared $\boldsymbol{C}$ in equation (3) with $\Re^2$ in equation (1), the search space of our hybrid approach is just composed by several points, which is much smaller than that of the direct approaches. Hence, it is time efficient. The above idea is used in horizontal and vertical vanishing point detection.

If the number of detected lines is enough, sufficient line intersections will be generated. And the true vanishing point will be embedded in these intersections with a high probability. Different from the conventional methods, our method finds the line candidates not only by paper boundaries lines but also by text baselines and the nature-born long character strokes in the text content. Therefore, the robustness of our method is improved.

3. Line and stroke detection and selection

When a document is lined up in a horizontal direction, we call it a horizontal document. Each text row has a clue horizontal direction. Our method uses document boundaries and text rows to detect the horizontal vanishing point. However, in the vertical direction of a horizontal document, there will be no text columns for vertical clues. Similar with the method described in [7], we extract the vertical character strokes as illusory vertical lines to detect a stable vertical vanishing point.

In Section 3.1, straight line detectin includes document boundaries and other straight lines. And text baseline detection is described in Section 3.2. Character tilt orientations are detected in Section 3.3.

3.1. Straight line detection

Line detection is solved with well-known conventional techniques. A lot of related work for such geometric problems is proposed, such as RANSAC-based methods, Least-Square-based methods, and Hough transform-based methods [14]. In this paper, in order to perform in a more efficient way, our line detection algorithms are based on edge information, connected component analysis, heuristic rules, and statistical analysis.

First, the input image is down-sampled. Then the edge is extracted by Canny edge detector [1]. Connected component analysis is used to find long connected components, which are merged in the horizontal or vertical direction according to shape and size information.

The merged connected components are regarded as line candidates. Given a connected component C_i, its corresponding line[2], LC_i, is fitted by the Least-Square algorithm.

The distance from one point (x,y) in C_i to the fitted line is

$$DIS_i(x,y) = \frac{|a_i y + b_i x + c|}{\sqrt{a^2 + b^2}}.$$

And we can get

$$f(LC_i) = \begin{cases} 1 & Len(LC_i) > len\_thres,\ N_{LC_i} > n\_thres\_line \\ 0 & otherwise \end{cases},$$

where

$$P_{LC_i}(x,y) = N(DIS_i(x,y), \mu_{line}, \sigma_{line}),$$

$$I_{LC_i}(x,y) = \begin{cases} 1 & P_{LC_i}(x,y) > p\_thres\_line \\ 0 & otherwise \end{cases},$$

and

$$N_{LC_i} = \sum_{(x,y)\in C_i} I_{LC_i}(x,y).$$

In the above equation, $Len(C_i)$ is the length of C_i, and $N(x,\mu,\sigma)$ is a Gaussian distribution for LC with mean μ and standard deviation σ. And μ_{line} and σ_{line} have been determined experimentally from different images.

If $f(LC_i)$ is equal to 1, then C_i will be a straight line.

Horizontal and vertical line detection and selection can be performed by the above steps respectively.

3.2. Horizontal text line smearing and detection

This process is based on a binarized image. We use a Block-Otsu algorithm for image binarization. After connected component analysis on the binary image, character candidates are selected by component size and shape analysis. Then, they are merged into horizontal text lines by a smearing algorithm derived from [5]. Finally, the horizontal direction of each smeared text line is computed.

The above procedure sometimes will produce smeared blocks that include more than one horizontal text lines because of perspective distortions. Therefore, we use a robust line direction detection method which is described in following.

First, we estimate the shape and size of each smeared text lines. Through vertical projection, we can obtain the upper and lower contour points of each smeared text line respectively. The upper contour points are

$$\{(x_1, y_1^U), (x_2, y_2^U), \ldots, (x_N, y_N^U)\},$$

where N is the width of the smeared text line. The lower contour points are

$$\{(x_1, y_1^L), (x_2, y_2^L), \ldots, (x_N, y_N^L)\}.$$

The average distance between each upper contour point and its corresponding lower contour point, *contour_thres*, is

[2] In this paper, the equation of lines is described as $ay+bx+c=0$.

then calculated.

If the distance of one contour point is less than *contour_thres*, then it is discarded. The reserved contour points are

$$Set(U) = \{(x_1, y_1^U), (x_2, y_2^U), ..., (x_M, y_M^U)\}$$

And

$$Set(L) = \{(x_1, y_1^L), (x_2, y_2^L), ..., (x_M, y_M^L)\},$$

where *M* is the number of the reserved contour points. And the middle points of the above contour points are

$$Set(C) = \{(x_1, (y_1^U + y_1^L)/2), (x_2, (y_2^U + y_2^L)/2), ..., (x_M, (y_M^U + y_M^L)/2)\}.$$

Three lines are fitted by the Least-Square algorithm according to the above upper, lower and middle contour points respectively: “the upper baseline”, “the lower baseline”, and “the center baseline”.

We select a smeared line as a real horizontal line when

$$cross\_angle(U, L) < angle\_thres$$

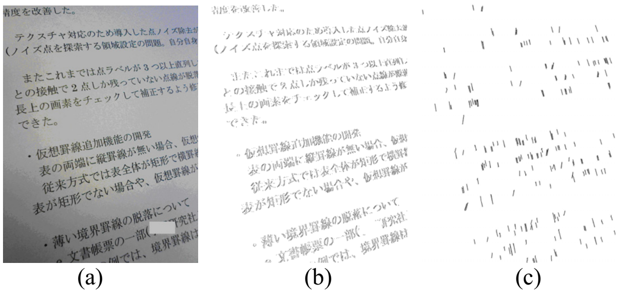
And

$$ave\_height(U, L) < height\_thres,$$

where *U* and *L* represent the upper and lower baselines respectively, *cross_angle* is the cross angle between two lines, and *ave_height* is the average height between the upper and lower baselines. Both *angle_thres* and *height_thres* are thresholds. And the horizontal direction of one text line is the direction of “the center baseline”.

3.3. Character vertical stroke extraction

In many situations, vertical clues are scarce. When an image is a partial portion of a whole document, there may be few or even no straight vertical lines. But character tilt orientations can be regarded as clue directions and the character vertical strokes can be used as vertical clues. However, these vertical clues are not stable. Though vertical stroke detection is solved with several conventional techniques [7], our method is rather efficient for MobileCam-based document analysis. Different from the multiple fuzzy sets used in [7], our method extracts a stable vertical stroke set by heuristic rules and statistical analysis.



(a) (b) (c)

Fig. 2. Vertical character stroke detection: (a) captured image, (b) edge image, (c) vertical strokes detected.

The character vertical stroke extraction is also based on the edge image of the document. Fig. 2(a) is the original perspective document, and Fig. 2(b) shows an example edge image. After connected components analysis on the vertical edge image, some long-stroke-like connected components are selected.

These selected connected components include vertical, horizontal, and slant direction. On the assumption that the perspective angle in the vertical direction is less than *45* degree, we select vertical stroke candidates by a simple rule: if the height of a connected component is much longer than its width, it is a vertical stroke candidate. In Chinese, Japanese and English characters and texts, there are many curve strokes, such as “八”, we remove these curve candidates by detecting the “straightness” of the stroke which is similar to the one described in Section 3.1.

Given a connected component C_i, its fitted line LC_i, and the distance from one point (x,y) in C_i to LC_i is $DIS_i(x,y)$, there is

$$f(LC_i) = \begin{cases} 1 & N_{LC_i} > n\_thres\_stroke \\ 0 & otherwise \end{cases},$$

where,

$$P_{LC_i}(x, y) = N(DIS_i(x, y), \mu_{stroke}, \sigma_{stroke}),$$

$$I_{LC_i}(x, y) = \begin{cases} 1 & P_{LC_i}(x,y) > p\_thres\_stroke \\ 0 & otherwise \end{cases},$$

and

$$N_{LC_i} = \sum_{(x,y) \in C_i} I_{LC_i}(x, y).$$

In the above equation, $N(x, \mu, \sigma)$ is a Gaussian distribution for LC with mean μ and standard deviation σ . And μ_{stroke} and σ_{stroke} are also determined experimentally. Because there are some noise and curves, we use the above steps to measure straightness of detected lines. That is, if one line is straight enough, it can be taken as a real line.

If $f(LC_i)$ is equal to 1, then C_i will be a vertical stroke. In order to detect straight vertical strokes, *p_thres_stroke* takes a high value (e.g., near to 1), and *n_thres_stroke* is near to the number of pixels in this component. The resulting vertical strokes of one document (Fig. 2(a)) are shown in Fig. 2(c).

Since in Chinese, Japanese and English texts, most slant strokes are curve strokes, after the above processing, the real vertical strokes are obtained. Consequently, the vertical vanishing point detection will be very robust.

4. Vanishing point detection by a hybrid approach

We use a hybrid approach for vanishing point detection. After the line intersections are calculated by line pairs, the intersection points are partitioned by clustering algorithm, and typical points are selected as reliable vanishing point candidates. This process can be viewed as an indirect approach. Next, a direct approach is performed by projection analysis from perspective views onto these point candidates. Finally, results of both approaches are linearly combined. The optimal candidate is selected as the final vanishing point.

4.1. Clustering for vanishing point detection

Without loss of generality, we describe the clustering based method for locating the horizontal vanishing point.

All horizontal lines (including straight lines and smeared lines detected in Section 3) are

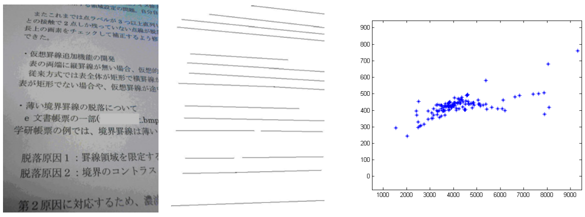
$$Set(Line)=\{(a_1,b_1,c_1),...,(a_N,b_N,c_N)\},$$

where N is the number of all horizontal lines.

As we know, two lines will produce an intersection point. As a result, there are $N_P = N\times(N-1)/2$ intersections which are possible candidates of the horizontal vanishing point. These intersections are

$$Set(Pt)=\{(x_1,y_1),...,(x_{N_P},y_{N_P})\}.$$

After checking the distribution of line intersections, we discover that these intersections are located in the 2D space with one or more groups with high density. It is natural to partition these points into groups by clustering. A sample of the distribution of intersections for a horizontal vanishing point is described in Fig. 3.



(a) (b) (c)

Fig. 3. Intersection distribution for a horizontal vanishing point: (a) captured image, (b) horizontal lines, (c) point distribution.

Our clustering space is 2D Euclidean space, and the similarity measure of two points is the Euclidean distance,

$$d(x_i,x_j)=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2},$$

where (x_i,y_i) is the feature vector (a point in the space).

The k-means clustering algorithm is a rather robust clustering algorithm, which is also proved from our initial experiences. The number of clusters in k-means is specified by the following simple rule:

$$N_{cluster}=\max(\lceil \ln(N_P)\rceil,\ 10).$$

4.2. Vanishing point detection and selection

For horizontal vanishing point detection, the first task is to locate several vanishing point candidates based on clustering introduced in Section 4.1. After clustering, we will obtain $N_{cluster}$ clusters. Each cluster is composed by some intersection points. The number of points in each cluster is

$$Set(Num)=\{N_1,N_2,...,N_{N_{cluster}}\}.$$

And the series of centers of all clusters is

$$X_C=\{x_1^c,x_2^c,...,x_{N_{cluster}}^c\},$$

where the center of each cluster is the average of all the intersection points in this cluster.

In our method, these centers are candidates of the horizontal vanishing point. And each candidate has a weight from clustering. The weight is given by

$$w_i^c=N_i/\sum_{i=1}^{N_{cluster}}N_i.$$

Each of these weights can be regarded as the profit function for the indirect approach. that is

$$f_{indirect}(x_i,y_i)=w_i^c(x_i,y_i).$$

In order to get a more stable vanishing point, we use a direct approach to refine vanishing point candidates in the above search space.

As shown in [2], for a perspectively skewed target, the bins represent angular slices projected from $\boldsymbol{H}(x,y)$, and mapping from an image pixel $\boldsymbol{p}$ for a bin $B_i(x,y)$ is as follows:

$$i(p,H)=\frac{1}{2}N+N\frac{\angle(H,H-p)}{\Delta\theta} \qquad (4)$$

where $\angle(H,H-p)$ is the angle between pixel $\boldsymbol{p}$ and the center of the image, relative to the vanishing point $\boldsymbol{H}(x,y)$, and $\Delta\theta$ is the size of the angular range within the document is contained, again relative to $\boldsymbol{H}(x,y)$. $\Delta\theta$ is obtained from

$$\Delta\theta=\angle(T_L,T_R)$$

where T_L and T_R are the two points on the bounding circle whose tangents pass through $\boldsymbol{H}(x,y)$. These are shown in Fig. 4.

For each cluster center, the above projection analysis is performed, and the derivative-squared-sum of the projection profiles from a perspective view is calculated,

$$f'_{direct}(c_i(x,y))=\sum_{j=1}^{N_B-1}(B_{j+1}(x,y)-B_j(x,y))^2. \qquad (5)$$

where $B(x,y)$ is a projection profile with a vanishing point candidate $\boldsymbol{H}(x,y)$, and N_B is the number of projection bins. This is the profit function for the direct approach.

For a computational convenience, the used profit is changed to a coefficient by

$$f_{direct}(c_i(x,y))=\frac{f'_{direct}(c_i(x,y))}{\sum_{i=1}^{N}f'_{direct}(c_i(x,y))}.$$

Then according to equation (2), we combine these two profits in a linear way,

$$g(x_i,y_i)=\alpha f_{indirect}(x_i,y_i)+\beta f_{direct}(x_i,y_i).$$

where $\alpha+\beta=1$. In our experiments, $\alpha=\beta=0.5$.

The resulting horizontal vanishing point is given by

$$(v_x,v_y)=\arg\max_{(x_i,y_i)\in X_C} g(x_i,y_i).$$

The last step is to confirm the resulting vanishing point. Our rejection strategy is that the derivative-squared-sum of the true vanishing point will be larger than values of other points. The derivative-squared-sum of the resulting vanishing point is $f'_{direct}(v_x,v_y)$, which is calculated by equation (5). The unchanged horizontal vanishing point is $(-\infty,0)$. And the derivative-squared-sum of it is $f'_{direct}(-\infty,0)$. If the following condition is satisfied, then the final horizontal vanishing point is (v_x, v_y):

$$f'_{direct}(v_x, v_y) > (1+\varepsilon) f'_{direct}(-\infty,0),$$

where $0 < \varepsilon < 1$, and in our method, $\varepsilon = 0.1$. Otherwise, we take a rejection strategy, and the final vanishing point will be $(-\infty,0)$.

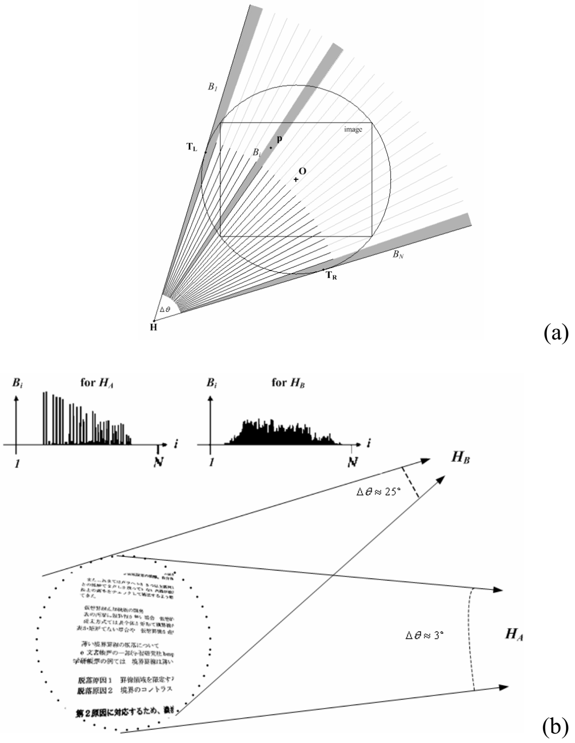


Fig. 4. Analysis of projection profiles from a perspective view: (a) generating projection profiles from $\boldsymbol{H}(x,y)$, (b) the winning projection profiles from the vanishing point $\boldsymbol{H}_A$ and a poor example from $\boldsymbol{H}_B$ (derived from [2]).

The way for the vertical vanishing point detection and selection is similar to the horizontal vanishing point. In vertical vanishing point detection, we use a projection analysis method which is slightly different from the one used in [2].

In character segmentation, vertical white-space often serves to separate different characters. In a similar way, given a vertical vanishing point candidate $\boldsymbol{V}(x,y)$, the bins represent angular slices projection from $\boldsymbol{V}(x,y)$ of each text row. Similar with equation (4), mapping from an image pixel $\boldsymbol{p}$ in the *kth* text row for the bin $B_i^k(x,y)$ is as follows:

$$i(p,V,k) = \frac{1}{2}N_k + N_k \frac{\angle(V, V-p)}{\Delta\theta_k},$$

where N_k is the number of bins on the *kth* text row. And $\angle(V, V-p)$ is the angle between $\boldsymbol{p}$ and the center of the image, relative to the vanishing point $\boldsymbol{V}(x,y)$, and $\Delta\theta_k$ is the size of the angular range within the *kth* text row is contained. Then, the profit function of the optimization becomes

$$f_{direct}(x,y) = \sum_{k=1}^{K}\sum_{i=1}^{N_k}(I_i^k(x,y)),$$

where $I_i^k(x,y) = 1$ or 0, and K is the number of text rows. If $B_i^k(x,y)$=0, then $I_i^k(x,y)$=1; otherwise, $I_i^k(x,y)$=0.

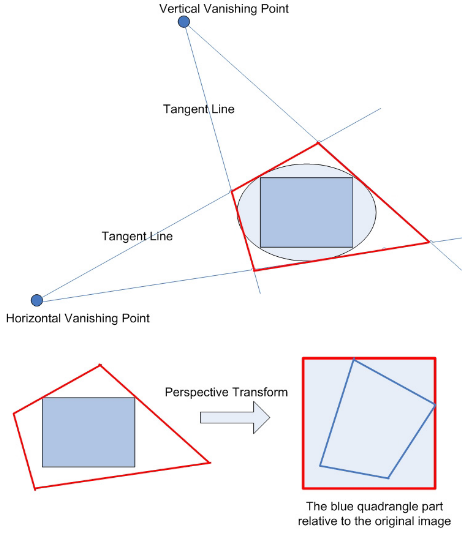
Consequently, a candidate with a maximum number of white columns of all rows from perspective views is the vertical vanishing point.

5. Experiments

The rectification transform of our system is performed as follows. Given the horizontal and vertical vanishing points, we can recovery documents with fronto-parallel views of the perspectively skew document. For perspective transformation on a *2D* plane, the transformation relation is

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix}$$

where (x_u, y_u) is the rectified (undistorted) image coordinate, and (x_d, y_d) is the original (distorted) image coordinate.



Vertical Vanishing Point
Tangent Line
Tangent Line
Horizontal Vanishing Point
Perspective Transform
The blue quadrangle part relative to the original image

Fig. 5. The perspective transform relation.

Given the horizontal and vertical vanishing points on the image plane, we can calculate a convex quadrangle on the image plane which is according to a rectangle in the rectified image plane. A versatile method for detecting such a convex quadrangle is described in Fig. 5. In Fig. 5, the ellipse is the circum-ellipse of the image rectangle.

The aspect ratio of the result image is decided as follows. The average length of the top and bottom sides of the quadrangle is the width of the result image, and the average length of the left and right sides of the quadrangle is the height of the result image.

The experiment database is *418* test samples captured by several mobile phone cameras. These images are in RGB color format with a 1280×960 resolution. More than *90%* of the images have perspective distortions, and other images have weak perspective distortions. Some samples are shown in Fig. 6.

Given a resulting vanishing point, VP, the relative distance from the ground truth VP_t is calculated. If

$$\frac{|VP - VP_t|}{|VP_t|} < T_{VP}, \qquad (6)$$

then VP is regarded as a correct vanishing point. In our system, the ground truth vanishing points are calculated from the manually marked horizontal and vertical lines. When the difference in Equation 6 is less than the threshold (T_{VP}=*1/20*), then there is no seemingly perspective distortion. It is also a conclusion from our experiments.

We divide our rectified images into five groups: (1) "**HIT**", successful for perspective rectification in both horizontal and vertical directions; (2) "**HHIT**", successful in the horizontal direction; (3) "**VHIT**", successful in the vertical direction; (4) "**REJ**", the rectified image is the same as the original image; (5) "**ERR**", represents rectifying with wrong perspective angles.

We compared our method (**M1**) to other four methods: **M2** doesn't use character vertical strokes for vertical vanishing point detection; **M3** uses the indirect approach based on clustering only to detect vanishing points; **M4** uses model fitting in [9] instead of clustering; and **M5** uses a sequential correction style (horizontal direction correction then vertical direction correction) to compare the speed with our method. The accuracy results are described in Fig. 7. And some rectified images with front-parallel views of our method are shown in Fig. 6. And the resulting image is the inner rectangle area of the detected perspective quadrangle.

As shown in Fig. 6, test samples include many different types. There are even some street signs and non-document images. The fraction of these non-document images is about *20%*. The "**REJ**" rate of our method is *26.32%*, which is mainly caused by too large distortions. For a mobile phone with some proper interactive GUIs, users may accept the results of **HIT**, **HHIT**, **VHIT**, and **REJ** because the resulting image from these has a much better quality than (or a same quality as) the originally captured image. In this way, the acceptance rate of our method is *98.33%*.

Compared with M2, our method (M1) improves the "HIT" groups by 11.48%. This shows that character vertical strokes are very useful to detect the vertical vanishing point for documents without vertical boundaries. Compared with M3, our method improves 6.94% for the "HIT" accuracy. Our hybrid approach is more adaptive and robust for vanishing point detection. Compared with M4, our method improves the "HIT" accuracy by 2.39% and decreases the processing time by 12ms, which shows that our clustering strategy is robust and fast compared to the traditional model fitting. Our method has a similar performance with M5, though M5 uses a sequential style with partial rectification.

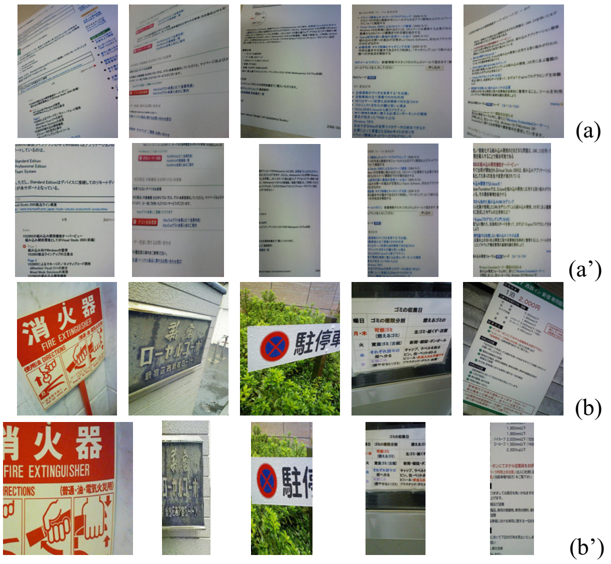


Fig. 6. Samples and the rectified images by our method (*x*' are the corresponding rectified images): (a) general documents, and (b) signboards and posters.

The processing speed is shown in Table 1, where "Time" represents the average processing time for each image without including the time for the grayscale image conversion and the final perspective transformation. Experiments are run on a DELL PC with 3GHz CPU, 2G Memory on a Windows XP OS.

Table 1. Results of the average processing time.

	M1	**M2**	**M3**	**M4**	**M5**
Time (ms)	103	72	90	115	226

As shown in Table 1, the average processing time of our method is largely less than M5, and the reduced time is 123ms. The additional time of M5 is spent in partially rectifying. We also test the direct approach by a hierarchical search for horizontal vanishing point detection in [2], which is more time consuming. For one image in test samples, the detection time is more than one second.

In conclusion, the acceptance rate of our method is more than *98%*, while the error rate is less than *2%*. And the processing time is only about *100*ms. With serious or unstable distortions, we take the rejection strategy, which may be more acceptable for a mobile user. Furthermore, with illusory lines derived from smeared text baselines and character tilt orientations, our method can rectify perspective documents without full boundaries (see Fig. 6(a)). All these show that our rectification method is fast and robust.

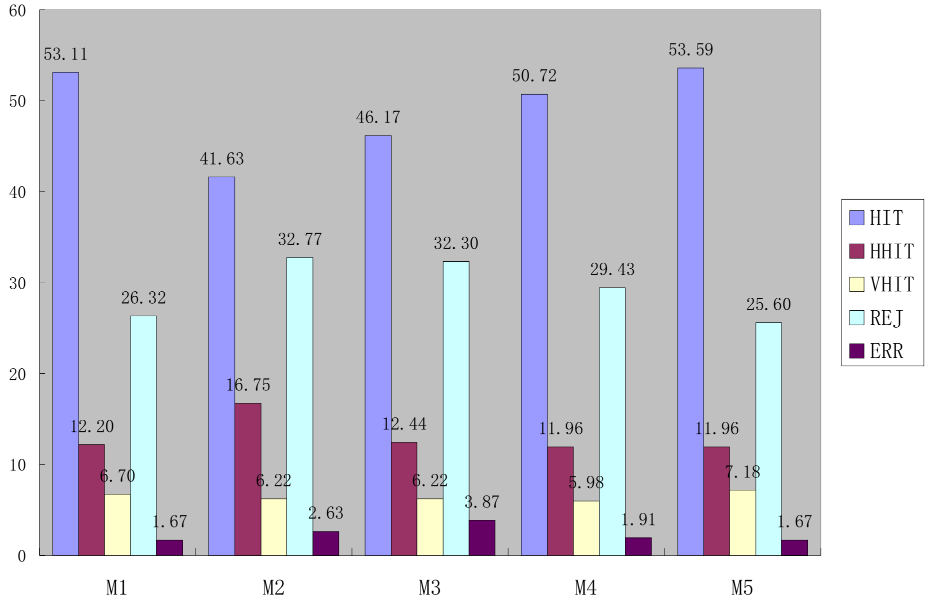


Fig. 7. Accuracy (%) of each resulting group.

6. Conclusions

Perspective rectification of MobileCam based documents faces several challenges, such as speed, robustness, non-boundary documents, etc. In this paper, we present a fast and robust method to deal with these problems. In our method, the hybrid approach for vanishing point detection combines direct and indirect approaches with high precision and fast speed. Furthermore, our method robustly detects text baselines and character tilt orientations by heuristic rules and statistical analysis, which is effective for documents without full boundaries. The experiments on different document images captured by mobile phone cameras show that our method has a good performance with an average speed of about *100*ms on a regular PC.

References

[1] J.F. Canny, "A computational approach to edge detection," *IEEE Trans. on PAMI*, vol. 8, no. 6, pp. 679-698, 1986.

[2] P. Clark, and M. Mirmehdi, "Rectifying perspective views of text in 3D scenes using vanishing points," *Pattern Recognition*, vol. 36, no. 11, pp. 2673-2686, 2003.

[3] C.R. Dance, "Perspective estimation for document images," *Proceedings of SPIE Conference on Document Recognition and Retrieval IX*, pp. 244-254, 2002.

[4] W.L. Hwang, C.-S. Lu, and P.-C. Chung, "Shape from texture: estimation of planar surface orientation through the ridge surfaces of continuous Wavelet transform," *IEEE Trans. on Image Processing*, vol. 7, no. 5, pp. 773-780, 1998.

[5] D.X. Le, G.R. Thoma, and H. Wechsler, "Automated borders detection and adaptive segmentation for binary document images," *Proceedings of International Conference on Pattern Recognition*, vol. 3, pp. 737-741, 1996.

[6] J. Liang, D. Doermann, and H.P. Li, "Camera-based analysis of text and documents: a survey," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2-3, pp. 84-104, 2005.

[7] S.J. Lu, B.M. Chen, and C.C. Ko, "Perspective rectification of document images using fuzzy set and morphological operations," *Image and Vision Computing*, vol. 23, no. 5, pp. 541-553, 2005.

[8] C. Monnier, V. Ablavsky, S. Holden, and M. Snorrason, "Sequential correction of perspective warp in camera-based documents," *Proceedings of IEEE Conference on Document Analysis and Recognition*, vol. 1, pp. 394-398, 2005.

[9] M. Pilu, "Extraction of illusory linear clues in perspectively skewed documents," *Proceedings of IEEE CVPR*, pp. 363-368, 2001.

[10] S. Pollard, and M. Pilu, "Building cameras for capturing documents," *International Journal on Document Analysis and Recognition*, vol. 7, no. 2-3, pp. 123-137, 2005.

[11] J. Shufelt, "Performance evaluation and analysis of vanishing point detection techniques," *IEEE Trans. on PAMI*, vol. 21, no. 3, pp. 282-288, 1999.

[12] J. Liang, D. DeMenthon, and D. Doerman, "Flattening curved documents in images," *Proceedings of IEEE CVPR*, pp. 338-345, 2005.

[13] J. Liang, D. DeMenthon, and D. Doerman, "Camera-based document image mosaicing," *Proceedings of International Conference on Pattern Recognition*, pp. 476-479, 2006.

[14] K.R. Castleman, *Digital Image Processing*, Prentice Hall, 1996.

Usage of continuous skeletal image representation for document images de-warping.

Anton Masalovitch, Leonid Mestetskiy
Moscow State University, Moscow, Russia
anton_m@abbyy.com, *l.mest@ru.net*

Abstract

In this paper application of continuous skeletal image representation to documents' image de-warping is described. A novel technique is presented that allows to approximate deformation of interlinear spaces of image based on elements of image's skeleton that lie between the text lines. A method for approximation of whole image deformation as combination of single interlinear spaces deformations is proposed and representation of it in the form of 2-dimensional cubic Bezier patch is suggested. Experimental results for batch of deformed document images are given that compare recognition quality of images before and after de-warping process. These results prove efficiency of the proposed algorithm.

1. Introduction

All the modern OCR systems assume that text lines in a document are straight and horizontal while in real images they are not. Image can be deformed before recognition in various ways. For example, if a thick book is scanned, text lines on the scan may be wrapped near the spine of book. If a digital camera is used to retrieve the image instead of a scanner, the text lines may be still wrapped because of low-quality optics of digital cameras. One important example of such deformation is the rounding of an image on borders as result of barrel distortion. Moreover, several types of deformation could be applied to the same image, making it impossible to build a precise model of image deformation. This is how the task of image de-warping appears.

The approach proposed in this paper is based on the construction of outer skeletons of text images. The main idea of the proposed algorithm is based on the fact that it is easy to mark up long continuous branches that define interlinear spaces of the document in outer skeletons. We approximate such branches by cubic Bezier curves to find a specific deformation model of each interlinear space of the document. On the basis of a set of such interlinear spaces' approximations, the whole approximation of the document is built in the form of a 2-dimensional cubic Bezier patch. After all this work is completed, we can de-warp an image using obtained approximation of image deformation.

This work is an extension of the article [1]. In this paper new method of automatic search for interlinear branches of skeleton is described. Also iteration method of image deformation approximation adjustment is given.

To test our algorithm we compare recognition results for a batch of images before and after the de-warping process.

2. Existing solutions

Algorithm of automatic image de-warping is needed nowadays for automatic OCR systems. Plenty of algorithms for image deformation approximation appeared in the last several years (see for example [7-11]). Unfortunately, most of these algorithms have some disadvantages that make them unusable for commercial OCR systems.

Existing solutions can be divided to three approaches:

- First approach is to single out text lines by combining close black objects and then approximating each line shape using some characteristic points of line's black objects. For example, one can approximate text lines' shape by using middle points of black objects' bounding rectangles. Main disadvantage of this approach is that it is hard to define such characteristic points of black objects that can give a stable approximation of line shape.
- Second approach is to build a model of possible deformation of an image and then try to apply this model for a specific image. Main disadvantage of this method is that it is almost impossible to build a complete model of image deformation. And if such a model describes only one type of deformation, one

should make sure that the used model can be applied for processing the concrete image.

- Finally, the third approach is to describe some estimation of text lines' straightness and iteratively deform image to achieve a maximum possible straightness of text lines. Main disadvantage of this method is that it uses numerical computing, and therefore is time-consuming, while the results of the method are often unpredictable.

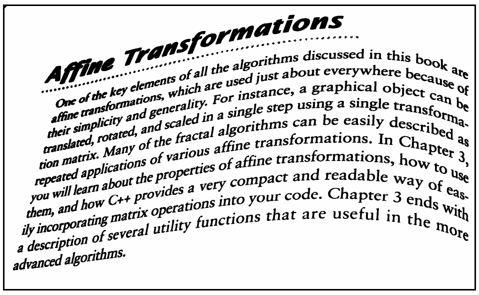
In our work we try to avoid described disadvantages. So our goal is to create image de-warping algorithm that does not depend on text symbols quality, is applicable to most of possible optic image deformations, with predictable results and not time-consuming.

3. Characteristics of images under consideration

It is necessary to describe some characteristics of images that our algorithm works with:

- Initial image should be black and white, with black text and white background. It should not contain inverted areas. It also should not contain noise or textures. In all modern OCR systems efficient add-ons exist that allow bringing almost every image to the marked model. And applied binarization and noise removal technique may be very rough because our algorithm does not depend on text symbols quality.
- Initial image should contain one big text block. This is an important assumption, because the proposed algorithm works with interlinear spaces rather than with text lines, and therefore initial image must contain a sufficient number of long text lines located one under another. All modern OCR systems can divide initial image into a set of text blocks with high precision, even when the images are deformed.
- Let us also assume that the deformation of text lines in an image can be approximated by continuous cubic curves and patches. This assumption is also not very restrictive, since most common deformations of images are created by cameras and scanners. Such deformations can be approximated even by quadratic patches and curves. As for more complicated cases, experiments have shown that cubic approximation is precise enough for them. In the case if additional experiments will show that cubic approximation is not sufficient after all, the degree of Bezier curves and patches can be easily increased without making considerable modifications to the proposed algorithm.

One example of an image with which our algorithm works is represented on figure 1.



Affine Transformations

One of the key elements of all the algorithms discussed in this book are affine transformations, which are used just about everywhere because of their simplicity and generality. For instance, a graphical object can be translated, rotated, and scaled in a single step using a single transformation matrix. Many of the fractal algorithms can be easily described as repeated applications of various affine transformations. In Chapter 3, you will learn about the properties of affine transformations, how to use them, and how C++ provides a very compact and readable way of easily incorporating matrix operations into your code. Chapter 3 ends with a description of several utility functions that are useful in the more advanced algorithms.

Figure 1. Processing image example

4. Problem definition

Let us assume that we have image $I(x,y)$, where I is the color of image pixel with coordinates (x,y). Let us also assume that this image contains text block with deformed lines. We further assume that we can rearrange pixels in this image without changing their colors to retrieve document image where initial lines become straight and horizontal. So, we want to develop a continuous vector function $\overline{D}(x,y)$ to obtain a de-warped image in the form: $I'(x,y) = I(\overline{D}_x(x,y); \overline{D}_y(x,y))$. This function $\overline{D}(x,y)$ will be an approximation of the whole image deformation.

To estimate the quality of our de-warping algorithm, we attempt to recognize the image before and after de-warping using one of the modern OCR systems. Recognition quality in modern OCR systems depends heavily on the straightness of text lines in images under consideration. Therefore, an improvement in recognition quality after image de-warping is a good evaluation of the quality of our de-warping algorithm.

5. Continuous border representation of binary image

In this work skeleton of polygonal figures is exploited. Before using such skeleton with binary images we must define representation of discrete binary image as a set of continuous polygonal figures.

Let us assume that a scanned document is stored in the form of a binary image represented as a Boolean matrix (one bit per pixel). A discrete model of the binary image is the integer lattice I in the Euclidean plane R^2 with 0 and 1 representing black and white elements. For elements of the lattice the relation of the 4-adjacent neighborhood is given. We designate $B \subset I$ as the set of black and $W \subset I$ as the set of white nodes of the lattice. Sets (B, W) serve as a model of the discrete binary image. In the same Euclidean plane R^2, we define the polygonal figure μ as the set of the points formed by association of a finite number of non-overlapping bounded closed domains. This figure is then a model of the continuous binary image. There is a problem consists in the construction of the figure μ that adequately describes properties of the discrete image B. In mathematical terms this problem is posed as an approximation of a discrete object with a continuous object. Natural criteria of good approximations should satisfy the following natural criteria:

1) $B \subset \mu$, $W \subset \left[R^2 \setminus \mu\right]$, where $[\ ]$ means closure of a set;

2) Let $x, y \in I$ be a pair of adjacent nodes of the lattice and s_{xy} be a segment connecting these nodes. Then if $x, y \in \mu$, then $s_{xy} \in \mu$, and if $x, y \notin \mu$ then $s_{xy} \cap \mu = \varnothing$.

The first condition means that the figure covers all black points of a discrete image and all white points lie either outside of or on the boundary of the figure. The second condition can be reduced to the condition that the boundary of μ lies in the interface between white and black boundary points of the discrete image.

Let M be the set of all figures satisfying conditions 1 and 2. Any of them can be considered a continuous model of a binary image with acceptable accuracy. As we are going to build a skeleton of this figure, the most convenient representation for us is the figure with a piecewise linear boundary, since for such figures there are effective algorithms for construction of a skeleton. In this situation it is natural to choose from M a polygonal figure (PF) with minimal perimeter (see fig. 2). First, such PF exists and it is unique. Second, the number of its vertices is close to minimal among all PF satisfying conditions 1 and 2.

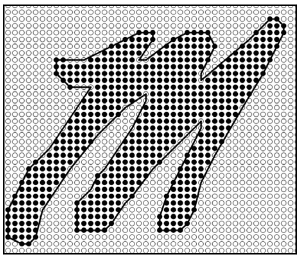


Figure 2. Representation of raster object with polygonal figure with minimal perimeter

The algorithm for solving this problem which requires a single pass over a raster image, has been described in [4].

6. Continuous skeletal representation of an image

The choice of the polygonal figure as a continuous model of the binary image reduces the problem of construction of a skeleton of the image to the well-known medial axis transform [5]. Contrary to discrete images for which the skeleton is determined ambiguously, the concept of a skeleton of a continuous figure has a strict mathematical formulation. The skeleton of a figure is the locus of points of centers of maximal empty circles. An empty circle does not contain any boundary points of the figure. The maximal empty circle is a circle which is not contained in any other empty circle, and which is not congruent to another. Note that empty circles can be thus either internal or external for the domains comprising the figure. Accordingly their centers form internal and external skeletons of the figure (see fig. 3).

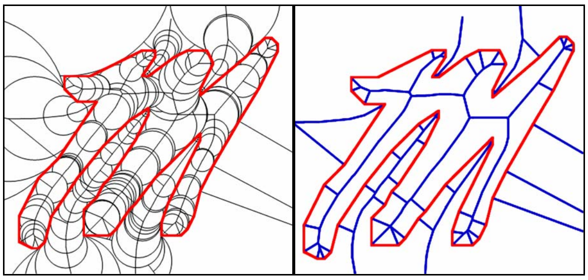


Figure 3. Empty circles for polygonal figure and skeleton of polygonal figure.

This definition applies to any type of shape, not just a polygon. However there exist effective algorithms for construction of polygonal figures [4,6]. The algorithm used [2,3] is based on a generalization of Delauney triangulation for a system of sites of two types (points and segments) that comprise a PF boundary. It builds a skeleton in time O(n log n) where n is the number of PF vertices.

Skeleton of polygonal figure can represented as a planar graph, where nodes are points on a plane and bones are straight lines that connect the nodes. In such representation of a skeleton all nodes have no less than three nearest points on the border of the area and all bones lie between two linear fragments of the area border. Later in this article we will use only graph representation of a skeleton.

Let us also define a knot in skeleton as a node with more then two connected bones and final node as a node with only one connected node. And let us define a branch of skeleton as a consistent set of bones that has final node or knot node on each end and does not have knots in the middle of the branch. Later in this article we will operate only with branches of the skeleton and not with single bones.

7. Main idea of the algorithm

Main idea of the proposed algorithm is that in outer skeleton of text document image, one can easily find branches that lie between adjacent text lines. Then, one can use this separation branches to approximate deformation of interlinear spaces in an image.

The proposed algorithm consists of the following steps:

- Continuous skeletal representation of an image is built.
- Skeleton is filtered (useless bones are deleted).
- Long near-horizontal branches of the skeleton are singled out.
- List of singled out branches is filtered to leave only branches that lie between different text lines.
- Cubic Bezier approximation is built for each branch.
- Bezier patch is built based on the obtained curves.

8. Image and skeleton preprocessing

As was mentioned before, one of the steps of our algorithm is the preprocessing step, on which we try to delete all small garbage branches and branches that can be obviously determined as non-interlinear from the skeleton. Let us describe this step in more detail.

First of all, before building a skeleton, we flood all white horizontal strokes with length smaller than some predefined threshold. By doing so, we glue symbols in words in one line, so we erase from image skeleton a lot if intersymbol branches that are useless for our algorithm. We set the value of the flooding parameter equal to 0.1 inches or 30 pixels for 300 dpi images (this value determined empirically). That value is sufficient to glue most adjacent symbols and not to glue adjacent curved lines.

Then we build outer skeleton of the expanded image.

The next step is to delete branches of the skeleton that divide different parts of the same object. Such branches describe borders of one symbol and are not relevant for the whole text line. We also delete branches of skeleton that divide objects in image and border of an image. Figure 4 shows an example of such image preprocessing.

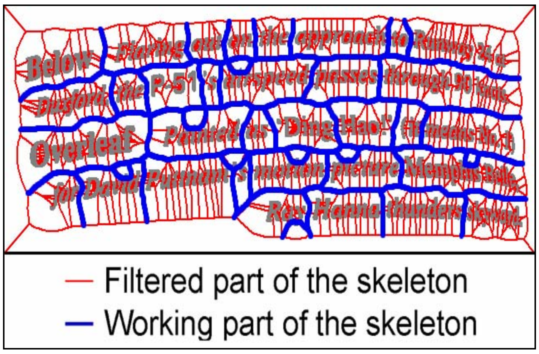

🟥 Filtered part of the skeleton
🟦 Working part of the skeleton

Figure 4. Image skeleton after preprocessing

9. Skeleton bones clusterization

After outer skeleton of a document image was built, we could divide branches of the skeleton into two groups: branches that lie between objects in one text line and branches that lie between adjacent text lines.

The main idea of the proposed algorithm is that such clusterization can be performed automatically for any document image.

First we sort out all skeleton branches that are shorter then some predefined threshold. Such branches appear when several long branches connected in one point. Such short branches works only for connectivity propose, the angle of such branches is unpredictable, so they are not used during clusterization process (see fig. 5).

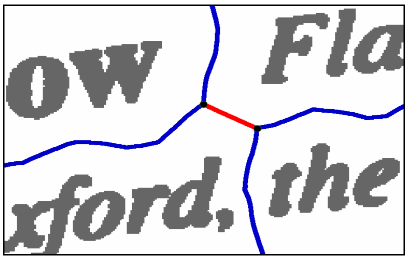


Figure 5. Short branch that connects several long brances.

As a threshold value for short branches we use empirical value of 0.05 inches or 15 pixels for 300 dpi images (determined empirically). It is about half of small letters height for standard font size, so we don't treat any of intersymbol branches as short.

To clusterize long branches we define parameter $A_{\max}$ - maximal absolute value of angle of interlinear branch (as angle of skeleton branch we use angle of linear approximation of that branch). Experiments show that it is possible for each image skeleton to define this parameter in such a way that all long vertical branches with |angle| > $A_{\max}$ will be only intersymbol branches.

This idea can be confirmed by graphic representation, if we draw all linear approximation of skeleton branches on one plane, so that they all begin in one point. For a document image the obtained figure will look like a cross, the horizontal part of which is created by interlinear branches, while the vertical part is created by intersymbol branches (see fig. 6).

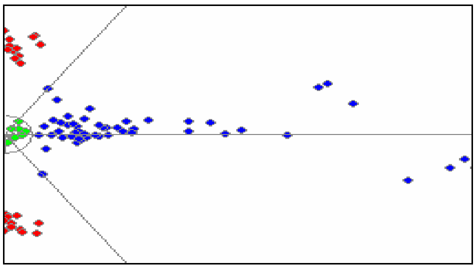


Figure 6. Branches of skeleton from figure 4 marked on one plane.

To define parameters $A_{\max}$ we use simple automatic clusterization mechanism. Each possible value of angle divides all branches into two classes with angle greater and less then the given threshold. For each class we define μ as the mean value of the angle in this class and σ as the standard deviation of the angle from the mean value. Using these two values we can define separation factor of two classes $J(t)$ in the form:

$$J(t)=\frac{\sigma_R+\sigma_L}{\mu_R-\mu_L}.$$

Then we iterate among the angles, looking for that with the minimum separation factor, using one degree as the size of the step (see fig. 7).

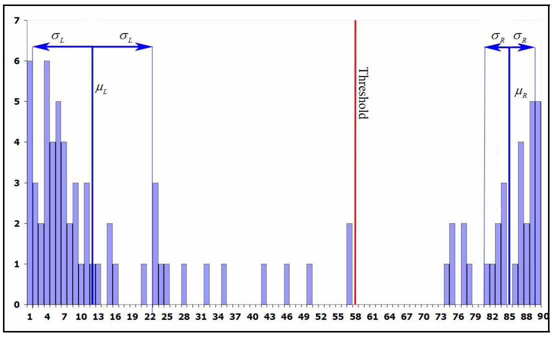


Figure 7. Histogramm of branches' angles from figure 6 with detected threshold

After clusterization we delete all long vertical branches from the skeleton (see fig. 8).

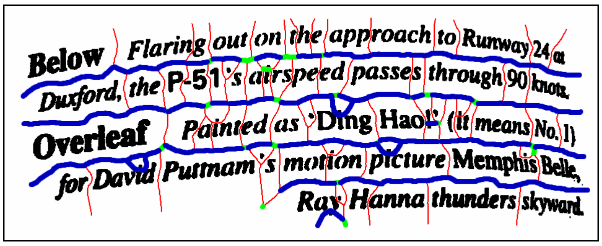

Below Flaring out on the approach to Runway 24 at
Duxford, the P-51's airspeed passes through 90 knots.
Overleaf Painted as 'Ding Hao!' (it means No. 1)
for David Putmam's motion picture Memphis Belle,
Ray Hanna thunders skyward.

Figure 8. Skeleton of document image after clusterization of branches.

10. Building interlinear branches.

After all vertical branches are deleted, the remaining branches are processed in cycle according to the following rules:

- If two nodes are connected by two non-intersected branches (such a problem appears when text language includes diacritics and additional branch goes between diacritic and symbol (see fig. 9)), we delete most curved branch of these two.

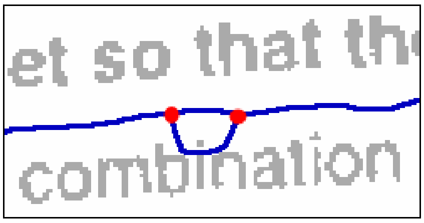


Figure 9. Two skeleton branches around a diacritic mark.

- If three branches are connected in one point (such a problem appears because some short branches remain after all vertical branches were deleted (see fig. 10)), we delete the shortest of the these branches.

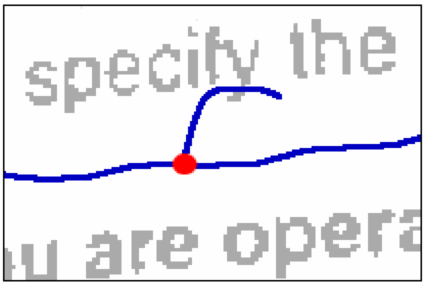


Figure 10. Remaining of vertical branches.

- If two long horizontal branches are connected near the border of an image (such a problem appears when two interline branches merge together outside the borders of a text block (see fig. 11)), we separate connection node of these branches into two independent nodes.

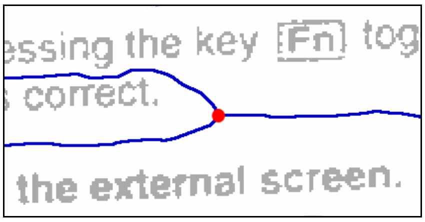


Figure 11. Two branches connected on the end of text line.

After all these rules were applied, only long horizontal branches that lie between adjacent text lines remain in the skeleton. We approximate them with cubic Bezier curves using method of least-square approximation.

11. Approximation of image deformation

After we get approximation of each interlinear space in the image, we must approximate deformation of whole image.

Define control points of interlinear curves as I_{ki}, where k is the index of a curve and i is the index of a control point on this curve.

For each set of points $\left\{\bigcup_{k=0}^{n} I_{ki}\right\}$ (control points from all interline curves with same index) we build approximation with vertical Bezier curve. Let us define control points of obtained curves as P_{ij}, where i is the index of initial control points and j is the index of new control points on created curve (see fig. 12).

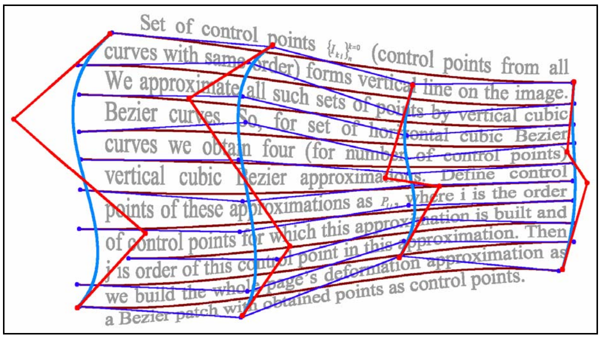

Set of control points $\{I_{k,i}\}_{k=0}^{n}$ (control points from all
curves with same order) forms vertical line on the image.
We approximate all such sets of points by vertical cubic
Bezier curves. So, for set of horizontal cubic Bezier
curves we obtain four (for number of control points)
vertical cubic Bezier approximations. Define control
points of these approximations as $P_{i,j}$, where i is the order
of control points for which this approximation is built and
j is order of this control point in this approximation. Then
we build the whole page's deformation approximation as
a Bezier patch with obtained points as control points.

Figure 12. Definition of control points of Bezier patch

After we get the set of points P_{ij}, we can build whole image deformation using Bezier patch. In other words, our approximation may be described by the following formula:

$$\overline{D}(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}P_{ij}*b_{i,3}(x)*b_{j,3}(y)$$

$b_{r,3}(t)$ - cubic Bernstein polynomial.

12. Bezier patch adjustment

Unfortunately, when we approximate interline spaces we cannot define clearily where each text line begins. Because of this, vertical points of the patch might be very randomly curved.

To avoid such an effect we use the following adjustment procedure:

For each interline curve $C_i(x)=\sum_{j=0}^{3}A_j^i*b_{j,3}(x)$ we search for nearest curve in Bezier patch. Define obtained curve as $\overline{C}_i(x)=\overline{D}(x,y_i)=\sum_{i=0}^{3}\sum_{j=0}^{3}P_{ij}*b_{i,3}(x)*b_{j,3}(y_i)$.

Define α and β as parameters of points on the curve $\overline{C}_i$ nearest to begin and end points of curve C_i.

$$\begin{cases}\alpha=\arg\min\limits_{t}\rho\left(\overline{C}_i(t),C_i(0)\right)\\ \beta=\arg\min\limits_{t}\rho\left(\overline{C}_i(t),C_i(1)\right)\end{cases}$$

Then we build curve C_i' that identical to C_i, but differs in parameterization (has shifted parameters), so that $C_i'(\alpha)=C_i(0)$ and $C_i'(\beta)=C_i(1)$. In other words,

$$C_i'(t)=C_i\left(t-\alpha\big/\beta-\alpha\right)=$$

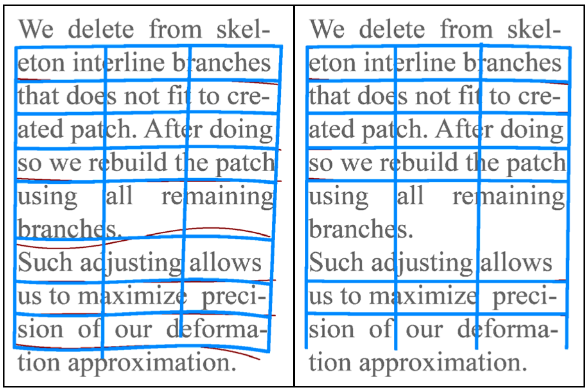
$$\sum_{j=0}^{3}A_j^i*b_{j,3}\left(t-\alpha\big/\beta-\alpha\right)=\sum_{j=0}^{3}B_j^i*b_{j,3}(t)$$

Then we calculate mean deviation d between curves C_i' and $\overline{C}_i$. If this deviation is greater than some predefined threshold, the original curve C_i must be excluded from patch creation, otherwise original curve C_i must be replaced with C_i'.

After the processing of all initial curves is completed we build a new Bezier patch using updated set of curves.

We repeat this procedure until deviations of all initial curves from curves from Bezier patch reach some predefined threshold.

This adjustment procedure allows to approximate vertical borders of text block and improves deformation approximation of whole page because of exclusion of erroneously created curves (see fig. 13).



We delete from skeleton interline branches that does not fit to created patch. After doing so we rebuild the patch using all remaining branches. Such adjusting allows us to maximize precision of our deformation approximation.

We delete from skeleton interline branches that does not fit to created patch. After doing so we rebuild the patch using all remaining branches. Such adjusting allows us to maximize precision of our deformation approximation.

Figure 13. Image deformation approximation before and after Bezier patch adjustment.

13. Experimental results

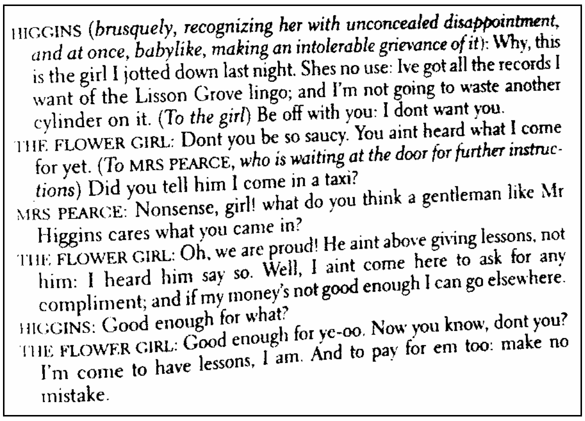
To test efficiency of our algorithm we take a set of 31 images. All images from this set satisfy the conditions described in section 3 – they are black-and-white images without noise, which contain one big text blocks with deformed lines. We recognize all these images with one modern OCR system before and after the de-warping process.

For deformed images there were 2721 recognition errors on all pages (4.92% of all). For de-warped images there were 830 recognition errors on all pages (1.50% of all symbols). Therefore, after the de-warping process 1891 errors were corrected (69.5% of original errors). In addition, 14 lines were not found on initial images, because of their high deformation, and after de-warping all text lines were defined correctly.

The attained results show high efficiency of the proposed algorithm, but its quality is not maximal yet. Recognition quality for straight images is higher than 99,5% in modern OCR systems. And for de-warped images we obtain the quality of only 98,5%. The main reason for this gap is that our algorithm deforms symbols a little during de-warping and that in turn causes errors in symbols' recognition.

On figures 14-16 an example of image de-warping for one of the images from our test set is given.

Also our algorithm was tested during Document Image De-warping Contest that was held in CBDAR 2007 [12]. On the contest de-warping algorithms were applied to test base of 100 images (test set available for download here - http://www.iupr.org/downloads/data). Experiments shown that mean edit distance for images de-warped by our algorithm was less then 1% on contest data set. Those results are statistically the same for the other two participants of the contest. And on quarter of test images our algorithm shown lowest edit distance.



HIGGINS (*brusquely, recognizing her with unconcealed disappointment, and at once, babylike, making an intolerable grievance of it*): Why, this is the girl I jotted down last night. Shes no use: Ive got all the records I want of the Lisson Grove lingo; and I'm not going to waste another cylinder on it. (*To the girl*) Be off with you: I dont want you.

THE FLOWER GIRL: Dont you be so saucy. You aint heard what I come for yet. (*To* MRS PEARCE, *who is waiting at the door for further instructions*) Did you tell him I come in a taxi?

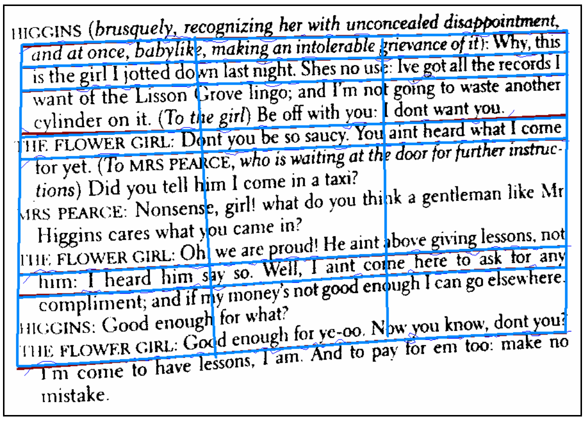
MRS PEARCE: Nonsense, girl! what do you think a gentleman like Mr Higgins cares what you came in?

THE FLOWER GIRL: Oh, we are proud! He aint above giving lessons, not him: I heard him say so. Well, I aint come here to ask for any compliment; and if my money's not good enough I can go elsewhere.

HIGGINS: Good enough for what?

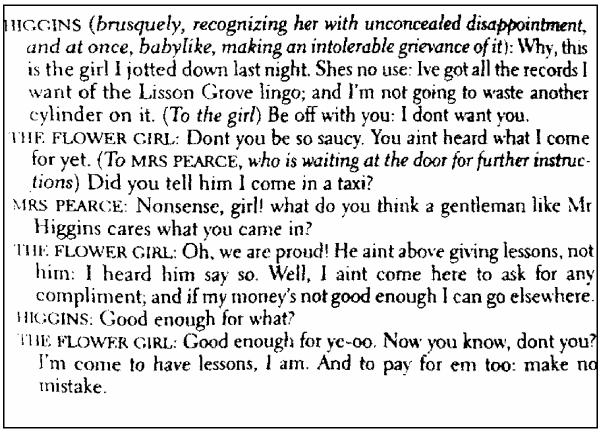
THE FLOWER GIRL: Good enough for ye-oo. Now you know, dont you? I'm come to have lessons, I am. And to pay for em too: make no mistake.

Figure 14. Initial deformed image



HIGGINS (*brusquely, recognizing her with unconcealed disappointment, and at once, babylike, making an intolerable grievance of it*): Why, this is the girl I jotted down last night. Shes no use: Ive got all the records I want of the Lisson Grove lingo; and I'm not going to waste another cylinder on it. (*To the girl*) Be off with you: I dont want you.
THE FLOWER GIRL: Dont you be so saucy. You aint heard what I come for yet. (*To* MRS PEARCE, *who is waiting at the door for further instructions*) Did you tell him I come in a taxi?
MRS PEARCE: Nonsense, girl! what do you think a gentleman like Mr Higgins cares what you came in?
THE FLOWER GIRL: Oh, we are proud! He aint above giving lessons, not him: I heard him say so. Well, I aint come here to ask for any compliment; and if my money's not good enough I can go elsewhere.
HIGGINS: Good enough for what?
THE FLOWER GIRL: Good enough for ye-oo. Now you know, dont you? I'm come to have lessons, I am. And to pay for em too: make no mistake.

Figure 15. Image deformation approximated with Bezier patch.



HIGGINS (*brusquely, recognizing her with unconcealed disappointment, and at once, babylike, making an intolerable grievance of it*): Why, this is the girl I jotted down last night. Shes no use: Ive got all the records I want of the Lisson Grove lingo; and I'm not going to waste another cylinder on it. (*To the girl*) Be off with you: I dont want you.
THE FLOWER GIRL: Dont you be so saucy. You aint heard what I come for yet. (*To* MRS PEARCE, *who is waiting at the door for further instructions*) Did you tell him I come in a taxi?
MRS PEARCE: Nonsense, girl! what do you think a gentleman like Mr Higgins cares what you came in?
THE FLOWER GIRL: Oh, we are proud! He aint above giving lessons, not him: I heard him say so. Well, I aint come here to ask for any compliment; and if my money's not good enough I can go elsewhere.
HIGGINS: Good enough for what?
THE FLOWER GIRL: Good enough for ye-oo. Now you know, dont you? I'm come to have lessons, I am. And to pay for em too: make no mistake.

Figure 16. De-warped image.

14. Future works

The main direction of feature work is to develop a better de-warping algorithm based on obtained image deformation approximation. De-warping algorithm that we use is very naïve, which heads to some additional recognition mistakes on the de-warped images.

More accurate approximation of deformation of vertical borders of text blocks is also one of our prior tasks.

15. Conclusion

This article describes a novel technique for approximation of text document image deformation based on continuous skeletal representation of an image.

In our work we try to avoid main disadvantages of existing de-warping solutions: use of separate symbol characteristics, use of specific deformation model, use of unpredictable numerical methods.

Main advantage of proposed algorithm is that it does not rely on quality of the initial text. Initial characters can be broken, flooded or erroneously binarized – proposed algorithm does not depend on it.

This paper describes all main steps of the proposed algorithm: construction of skeletal representation of an image, preprocessing of image's skeleton, detection of interlinear branches of the skeleton, approximation of such branches, final approximation of image deformation.

Based on the proposed algorithm a prototype of fully automatic system of image de-warping was built.

Experimental results that prove efficiency of the proposed algorithm and its importance for recognition of deformed images are given.

Acknowledgments

This research was supported by a grant from Russian foundation for basic research (grant 05-01-00542).

Bibliography

[1] A.A. Masalovitch, L.M. Mestetskiy, "Document Image Deformation Approximated by the Means of Continuous Skeletal Representation of the Image", Proceedings of international conference PRIP (Pattern Recognition and Information Processing), 2007, pp. 279-284.

[2] L.M. Mestetskiy, "Skeletonization of polygonal figures based on the generalized Delaunay triangulation", Programming and computer software, 25(3), 1999, pp. 131-142.

[3] L.M. Mestetskiy, "Skeleton of multiply connected polygonal figure", Proceedings of international conference "Graphicon", 2005.

[4] S. Fortune, "A sweepline algorithm for Voronoi diagrams", Algorithmica, 2, 1987, pp. 153-174.

[5] D.T. Lee, "Medial axes transform of planar shape", IEEE Trans. Patt. Anal. Mach. Intell. PAMI-4, 1982, pp.363-369.

[6] C.K. Yap, "An O(n log n) algorithm for the Voronoi diagram of the set of simple curve segments", Discrete Comput. Geom., 2, 1987, pp.365-393.

[7] Hironori Ezaki, Seiichi Uchida, Akira Asano, Hiroaki Sakoe, "Dewarping of document images by global optimization", Proceedings of international conference ICDAR, 2005, pp. 302-306.

[8] Udrian Ulges, Christoph H. Lampert, Thomas M. Breuel, "A Fast and Stable Approach for Restoration of Warped Document Images", Proceedings of international conference ICDAR, 2005, pp. 384-388.

[9] Li Zhang, Chew Lim Tan, "Warped Image Restoration with Application to Digital Libraries", Proceedings of international conference ICDAR, 2005, pp. 192- 196.

[10] A. Yamashita, A. Kawarago, T. Kaneko, K.T. Muira, "Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system", Proceedings of international conference ICPR, 2004, pp 482-485.

[11] M.S. Brown, W.B. Seales, "Image Restoration of Arbitrarily Warped Documents". IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26 , Issue 10, 2004, pp. 1295 - 1306.

[12] Faisal Shafait, Thomas M. Breuel, "Document Image Dewarping Contest". Proceedings of 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil, Sep. 2007.

Instance-Based Skew Estimation of Document Images by a Combination of Variant and Invariant

Seiichi Uchida
Kyushu Univ., Japan
uchida@is.kyushu-u.ac.jp

Megumi Sakai
Kyushu Univ., Japan
sakai@human.is.kyushu-u.ac.jp

Masakazu Iwamura
Osaka Pref. Univ., Japan
masa@cs.osakafu-u.ac.jp

Shinichiro Omachi
Tohoku Univ., Japan
machi@aso.ecei.tohoku.ac.jp

Koichi Kise
Osaka Pref. Univ., Japan
kise@cs.osakafu-u.ac.jp

Abstract

A novel technique for estimating geometric deformations is proposed and applied to document skew (i.e., rotation) estimation. The proposed method possesses two novel properties. First, the proposed method estimates the skew angles at individual connected components. Those skew angles are then voted to determine the skew angle of the entire document. Second, the proposed method is based on instance-based learning. Specifically, a rotation variant and a rotation invariant are learned, i.e., stored as instances for each character category, and referred for estimating the skew angle very efficiently. The result of a skew estimation experiment on 55 document images has shown that the skew angles of 54 document images were successfully estimated with errors smaller than 2.0 degree. The extension for estimating perspective deformation is also discussed for the application to camera-based OCR.

1. Introduction

For document image processing, the estimation of geometric deformations is an important problem. For example, many researchers have widely investigated the estimation of skew deformation, which severely degrades performance of OCR. Recently, the estimation of perspective deformation also becomes important problem toward the realization of camera-based OCR [1].

In this paper, a novel deformation estimation method is proposed and its performance is evaluated qualitatively and quantitatively via several experiments. In principle, the proposed method can estimate various geometric deformations such as perspective deformation and affine deformation. In this paper, we will focus on the document skew (i.e., rotation) estimation problem, which will be a reasonable problem to observe the basic performance of the proposed method.

The proposed method possesses two novel properties. First, the proposed method estimates the skew angles at individual connected components. (Note that each connected component may be a character). Those skew angles are then voted to determine the skew angle of the entire document. This fact implies that the proposed method does not rely on the straightness of text lines, whereas the conventional skew estimation methods totally rely on the straightness. Thus, the proposed method can be applied to documents whose component characters are laid out irregularly. This property is very favorable for camera-based OCR whose targets will often be short words and/or characters laid out freely.

Second, the proposed method employs instance-based learning for estimating the skew angle by referring stored instances. The simplest realization of instance-based skew estimation will be done by using font images as instances; the skew angle of each connected component is estimated by rotating and matching the font image to the connected component. The rotation angle giving the best match is the estimated skew angle. This simple realization, however, is very naive and requires huge computations. Specifically, it requires $O(N \cdot C \cdot K)$ image matchings, where N is the number of connected components in the target document, C is the number of instances (i.e., the number of assumed character categories), and K is the number of quantized angles (e.g., 360 for the estimation of 1° resolution). The proposed method avoids this problem by using a *rotation invariant* and a *rotation variant* as the instances and therefore requires only $O(N)$ (or less) computations.

The rest of this paper is organized as follows. After a brief review of the conventional methods in Section 2, the proposed method is described in Section 3. The role of the variant and the invariant is also detailed. Through these descriptions, it will be clarified that the proposed method does not rely on the straightness of text lines. In Section 4, the performance of the proposed method is observed via a skew estimation experiment of several document images. After remarking the extensibility of the proposed method for estimating deformations other than rotation in Section 5, a conclusion is drawn in Section 6 with a list of future work.

2. Related Work

Thee skew estimation strategies of the conventional methods are classified into two types, global estimation and local estimation. The global estimation strategy utilizes global features such as projection histogram, whereas the local estimation strategy utilizes local features such as the principal axis of adjacent connected components. In the latter strategy, local skew angles are estimated first by the local features and then combined to determine the skew angle of the entire document. Although the local estimation strategy is minority, it possesses several good properties. Especially, its robustness to irregular layouts (such as short or scattered text lines, figures, mathematical notations, and multi-column layouts). It also has the extensibility to non-uniform skew estimation problems such as document image dewarping and multi-skew detection [2].

In Ishitani [3], a local skew angle is estimated within a circular window on a document. This local skew estimation is done by searching for the direction which gives the most intensity transitions. Among the estimated skew angles at different windows, the most reliable one is chosen as the global skew angle. Jiang et al. [4] have employed a least mean square fitting for estimating local skew angles. They choose the global skew angle by voting those local skew angles. Lu and Tan [5] have determined a group of connected components (called a nearest-neighbor chain) which comprise a word (or a sub-word) by a region growing technique and then its skew angle is estimated. The global skew angle is chosen as the medium or the mean of the local skew angles. Lu and Tan [6] have proposed an interesting method which utilizes the straight strokes of individual characters for estimating local skew angles.

All of those conventional methods rely on the local straightness of the text lines and/or character strokes. The proposed method does not assume any straightness and thus possesses far more robustness to irregular layouts than the conventional methods. As noted in Section 1, this property is favorable for camera-based OCR.

3. Instance-Based Skew Estimation

3.1. Learning instance

The proposed method estimates the skew angle of each connected component in the target document by referring stored instances. The detail of the estimation will be discussed in Section 3.2. This section describes how to learn the instances, i.e., how to prepare the instances.

The instances are comprised of a rotation variant $p_c(\theta)$ and a rotation invariant q_c where $c \in [1, \ldots, C]$ denotes the character category[1] and θ denotes the skew angle. They are prepared according to the following steps:

1. Define the C character categories which will be included in target documents.
2. For each category c,
   (a) prepare the font image $\boldsymbol{R}_c$,
   (b) measure the value of the rotation invariant q_c, and
   (c) measure the value of the rotation variant $p_c(\theta)$ by rotating $\boldsymbol{R}_c$ by θ.

While any rotation variant and invariant can be used, the following simple variant and invariant are used as $p_c(\theta)$ and q_c in this paper:

$$p_c(\theta) = \frac{\text{area of bounding box of } \boldsymbol{R}_c \text{ at } \theta}{\text{area of black pixels of } \boldsymbol{R}_c}, \qquad (1)$$

$$q_c = \frac{\text{area of convex hull of } \boldsymbol{R}_c}{\text{area of black pixels of } \boldsymbol{R}_c}. \qquad (2)$$

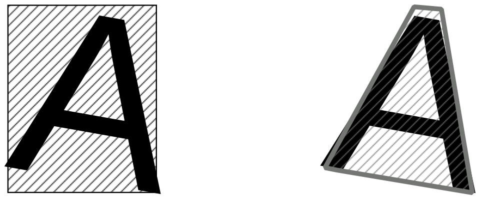
Fig. 1 shows the bounding box and the convex hull of a character. The area of the bounding box depends on the rotation angle θ, whereas the area of the convex hull does not. Fig. 2 shows the variant $p_c(\theta)$ of “y”(Times-Roman) as a function of θ. This function is stored as an instance together with q_c.

The rotation variant of (1) becomes a periodic function of $[-45^\circ, 45^\circ]$. Thus, the variant cannot distinguish, for example, 30° and 120°. If necessary, it is possible to avoid this periodic property by using a variant other than (1).

Note that both $p_c(\theta)$ and q_c are scale and shift invariants. (Thus, q_c is an invariant to similarity transformation.) This scale and shift invariance implies that the proposed method can estimate the correct skew angle regardless of the character size and position.

Although we should define the categories at the learning step, this definition need not to be strict; that is, the proposed method can estimate correct skew angle even if the

[1] Different fonts and styles belong to different categories. Thus, $C = 52 \times 3 \times 2 = 312$, when we assume three styles (e.g., “upright”, “italic”, and “bold”) and two fonts (e.g., “Times-Roman” and “Sans Serif”) for 52 categories of “A”∼“Z” and “a”∼“z.”



(a) (b)

Figure 1. (a) Bounding box and (b) convex hull of $c =$“A.”

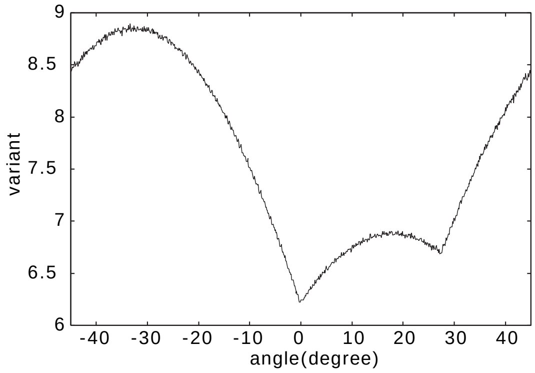


Figure 2. The variant $p_c(\theta)$ of “y.”

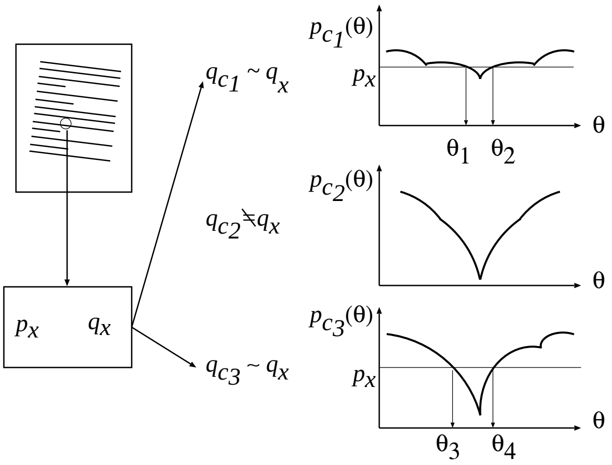


Figure 3. Estimation of skew angle by variant and invariant.

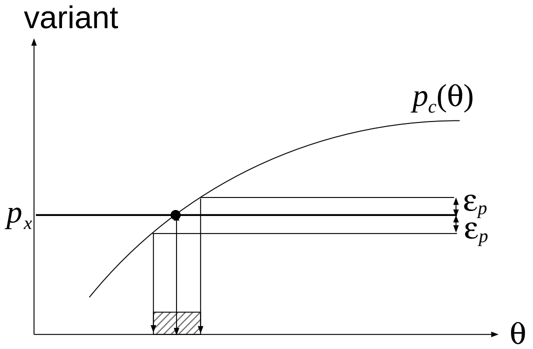


Figure 4. The range of the skew angle for variant p_x.

target document includes the characters whose instances are not learned. As discussed later, this robustness comes from the voting strategy for determining the skew angle of the entire document. In Section 4, the robustness will be experimentally shown through the skew estimation result of a mathematical document which includes several undefined characters (i.e., mathematical symbols).

3.2. Skew estimation by instances

The estimation of the skew angle of a binarized document image is done by a three-step manner: (i) the estimation of the category of each connected component by the invariant q_c (→3.2.1), (ii) the estimation of the skew angle of the connected component by the estimated category c and the variant $p_c(\theta)$ (→3.2.2), and (iii) the estimation of the skew angle of the entire document by voting (→3.2.3).

3.2.1 Category estimation by invariant

Let $\boldsymbol{X}$ denote a connected component ($\simeq$ a character) of the target document image. The category of $\boldsymbol{X}$ can be estimated by comparing its invariant value q_x to the stored instances $\{q_c | c = 1, \ldots, C\}$. If $q_c = q_x$, c is the estimated category of $\boldsymbol{X}$. Since q_x and q_c are rotation invariants, this estimated category will not change under any skew angle.

In practice, the connected component $\boldsymbol{X}$ may be contaminated by noises at image acquisition and thus q_x will be changed from its original value. Thus, all the categories satisfying $|q_c - q_x| \leq \epsilon_q$ are considered as the estimated categories, where ϵ_q is a non-negative constant. In the example of Fig. 3, we will obtain two estimated categories c_1 and c_3 because $q_x \sim q_{c_1} \sim q_{c_3}$, whereas $q_x \not\sim q_{c_2}$.

The category candidate c which satisfies $|q_c - q_x| \leq \epsilon_q$ can be found very efficiently by a look-up table which is indexed by the invariant value q_x. This efficiency is especially useful when a large number of categories are defined for dealing with various fonts, mathematical symbols, multilingual documents, and so on.

3.2.2 Local skew estimation by variant

For each estimated category c, the skew angle of $\boldsymbol{X}$ can be estimated by comparing its variant value p_x to the stored variance $p_c(\theta)$. The angle θ satisfying $p_c(\theta) = p_x$ is a candidate of the skew angle of $\boldsymbol{X}$ and therefore a candidate of the document skew angle. (Precisely speaking, we will use the relaxed condition $|p_c(\theta) - p_x| \leq \epsilon_p$ instead of $p_c(\theta) = p_x$, where ϵ_p is a non-negative constant. This will be discussed in 3.2.3.)

Consequently, multiple skew angle candidates will be obtained from a single connected component $\boldsymbol{X}$. This is because a single connected component $\boldsymbol{X}$ will have multiple category candidates and, furthermore, each category candidate c will provide multiple skew angle candidates satisfying $p_c(\theta) = p_x$. In the example of Fig. 3, two candidates θ_1, θ_2 are obtained from $p_{c_1}(\theta)$ and two candidates θ_3, θ_4 are from $p_{c_3}(\theta)$. In other words, four candidates are obtained from a single connected component.

Like the above category estimation step, the angle θ which satisfies $p_x = p_c(\theta)$ can be found very efficiently by a look-up table indexed by the variant value. This table is considered as the inverse function $\theta = p_c^{-1}(p_x)$.

3.2.3 Global skew estimation by voting

A voting strategy is employed for estimating the skew angle of the entire document. Roughly speaking, the purpose of the voting is to find the most frequent skew angle among all the candidates obtained by the above (local) skew estimation step. The voting strategy makes the proposed method tolerant to the false category candidates and the false skew angle candidates. Another merit is the tolerance to undefined categories. The bad effect of the undefined categories can be minimized by voting far more candidates representing the correct skew angle.

The skew angle of the entire document is estimated by voting the “range” specified by each skew angle candidate. As shown in Fig. 4, the range is determined as $[p_c^{-1}(p_x - \epsilon_p), p_c^{-1}(p_x + \epsilon_p)]$ by assuming that the true value of the variant p_x lies within $[p_x - \epsilon_p, p_x + \epsilon_p]$. (This range is come from the relaxed condition $|p_c(\theta) - p_x| \leq \epsilon_p$.) The skew angle is finally obtained as the angle where the most ranges are overlapped.

It is noteworthy that the width of the range is negatively proportional to the reliability of the skew angle candidate. This fact can be understood from the following example: Consider an “o”-shaped character. The reliability of the skew angle estimated at the character will be low because the character does not change its shape by any skew. In this case, its skew variant $p_c(\theta)$ will change subtly according to θ and the range determined by the variant becomes wide. In contrast, the variant of an “I”-shaped character, which will provide a highly reliable skew angle, will change drastically according to θ and the range becomes narrow.

3.2.4 Computational feasibility

The proposed method has a strong computational feasibility. This strength is emphasized not only by the use of the invariant and the variant but also by the look-up tables. The proposed method does not perform any try-and-error skew estimation step, unlike the global estimation methods based on the projection histogram and the local estimation methods like [3]. Furthermore, the proposed method requires neither line fitting nor image processing to search neighborhoods of each connected component. The proposed method, of course, does not require any costly image matching procedure, unlike the simple realization of the instance-based skew estimation outlined in Section 1.

The computational feasibility of the proposed method may be further improved by using a limited number of connected components. In fact, it is not necessary to use all the connected components in the document as experimentally shown later. This is because all the connected components, in principle, will show the same skew angle and thus the voting result will show the peak at the correct skew angle even with a limited number of votes.

4. Experimental Results

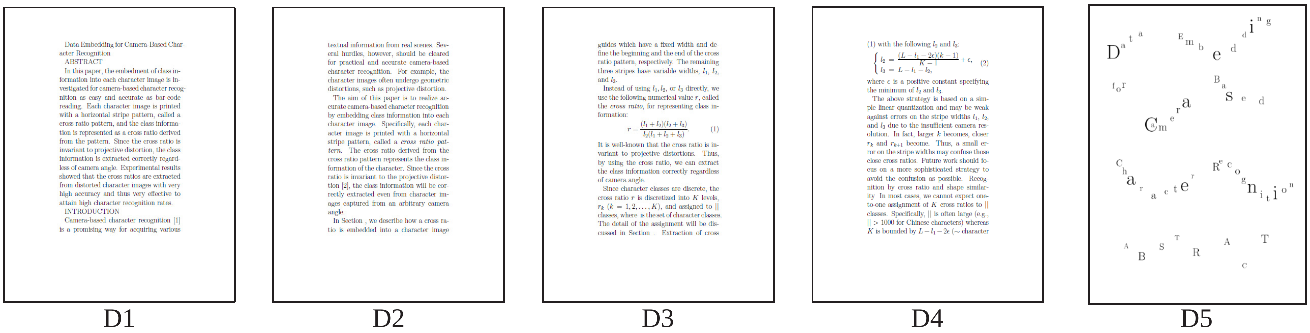
4.1. Document image samples

Five document images were created by LaTeX with a single font and style (Times-Roman, upright) and used for the evaluation of the skew estimation accuracy. Their resolution was 600 dpi. Fig. 5 shows those images, D1∼D5. The number of the defined categories were $C = 52$ (“A”∼ “Z”, “a”∼ “z”,). It is noteworthy that the two documents D3 and D4 include mathematical expressions and thus include several undefined categories, such as italic fonts and mathematical symbols. The document D5, where characters were freely laid out, was prepared to emphasize the robustness of the proposed method to irregular layouts.

Each document image was rotated $\pm 30°, \pm 20°, \pm 10°, \pm 5°, \pm 2°, 0°$ and thus 55 test images were prepared in total. Fig. 6 shows several rotated images of D1. For every connected component in the document image, its category and skew angle were estimated and voted to determine the skew angle of the entire document image.

4.2. Preparing instances

For each of the 52 categories, the instances, i.e., the variant $p_c(\theta)$ and the invariant q_c, were measured by using the original font image of Times-Roman as $\boldsymbol{R}_c$ and stored. The resolution of the font image was 1440 dpi. Both the variant



Data Embedding for Camera-Based Character Recognition

ABSTRACT

In this paper, the embedment of class information into each character image is investigated for camera-based character recognition as easy and accurate as bar-code reading. Each character image is printed with a horizontal stripe pattern, called a cross ratio pattern, and the class information is represented as a cross ratio derived from the pattern. Since the cross ratio is invariant to projective distortion, the class information is extracted correctly regardless of camera angle. Experimental results showed that the cross ratios are extracted from distorted character images with very high accuracy and thus very effective to attain high character recognition rates.

INTRODUCTION

Camera-based character recognition [1] is a promising way for acquiring various

textual information from real scenes. Several hurdles, however, should be cleared for practical and accurate camera-based character recognition. For example, the character images often undergo geometric distortions, such as projective distortion.

The aim of this paper is to realize accurate camera-based character recognition by embedding class information into each character image. Specifically, each character image is printed with a horizontal stripe pattern, called a *cross ratio pattern*. The cross ratio derived from the cross ratio pattern represents the class information of the character. Since the cross ratio is invariant to the projective distortion [2], the class information will be correctly extracted even from character images captured from an arbitrary camera angle.

In Section , we describe how a cross ratio is embedded into a character image

guides which have a fixed width and define the beginning and the end of the cross ratio pattern, respectively. The remaining three stripes have variable widths, l_1, l_2, and l_3.

Instead of using l_1, l_2, or l_3 directly, we use the following numerical value r, called the *cross ratio*, for representing class information:

$$r = \frac{(l_1 + l_2)(l_2 + l_3)}{l_2(l_1 + l_2 + l_3)}. \qquad (1)$$

It is well-known that the cross ratio is invariant to projective distortions. Thus, by using the cross ratio, we can extract the class information correctly regardless of camera angle.

Since character classes are discrete, the cross ratio r is discretized into K levels, r_k $(k = 1, 2, \ldots, K)$, and assigned to || classes, where is the set of character classes. The detail of the assignment will be discussed in Section . Extraction of cross

(1) with the following l_2 and l_3:

$$\begin{cases} l_2 = \dfrac{(L - l_1 - 2\epsilon)(k-1)}{K-1} + \epsilon, \\ l_3 = L - l_1 - l_2, \end{cases} \qquad (2)$$

where ϵ is a positive constant specifying the minimum of l_2 and l_3.

The above strategy is based on a simple linear quantization and may be weak against errors on the stripe widths l_1, l_2, and l_3 due to the insufficient camera resolution. In fact, larger k becomes, closer r_k and r_{k+1} become. Thus, a small error on the stripe widths may confuse those close cross ratios. Future work should focus on a more sophisticated strategy to avoid the confusion as possible. Recognition by cross ratio and shape similarity In most cases, we cannot expect one-to-one assignment of K cross ratios to || classes. Specifically, || is often large (e.g., || > 1000 for Chinese characters) whereas K is bounded by $L - l_1 - 2\epsilon$ ($\sim$ character

D a t a E m b e d d i n g
f o r B a s e d
C a m e r a
C h a r a c t e r R e c o g n i t i o n
A B S T R A C T
D1
D2
D3
D4
D5

Figure 5. Five document images used in the experiment. D3 and D4 include mathematical expressions.

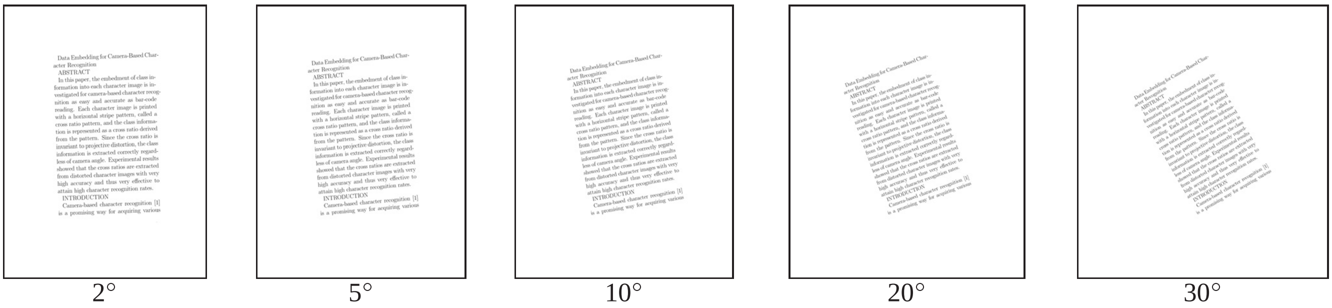
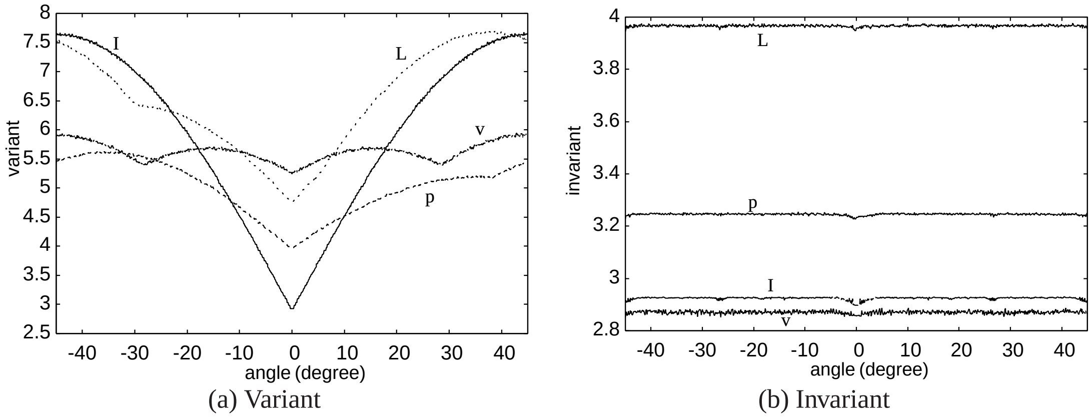


Figure 6. Skewed document images (D1).



(a) Variant

(b) Invariant

Figure 7. Examples of variant and invariant.

and the invariant were measured by rotating $\boldsymbol{R}_c$ every $0.1°$ from $-45°$ to $45°$. As noted before, the italic fonts and the mathematical symbols included in D3 and D4 did not have their own instances.

Fig. 7 (a) shows the variants of several categories. The variants $p_c(\theta)$ of "I" and "L" change drastically according to θ whereas the variant of "v" only changes subtly. As noted in 3.2.3, the variants changing drastically are favorable for the reliable skew estimation. In the experiment, however, the variants changing subtly were also used for observing the basic performance of the proposed method.

Fig. 7 (b) shows the invariants of several categories. The invariants, in principle, will not change according to θ. This figure, however, reveals that the invariant fluctuates due to noise at image acquisition. In the experiment, the invariant value was averaged from $-45°$ to $45°$ and then stored as the instance.

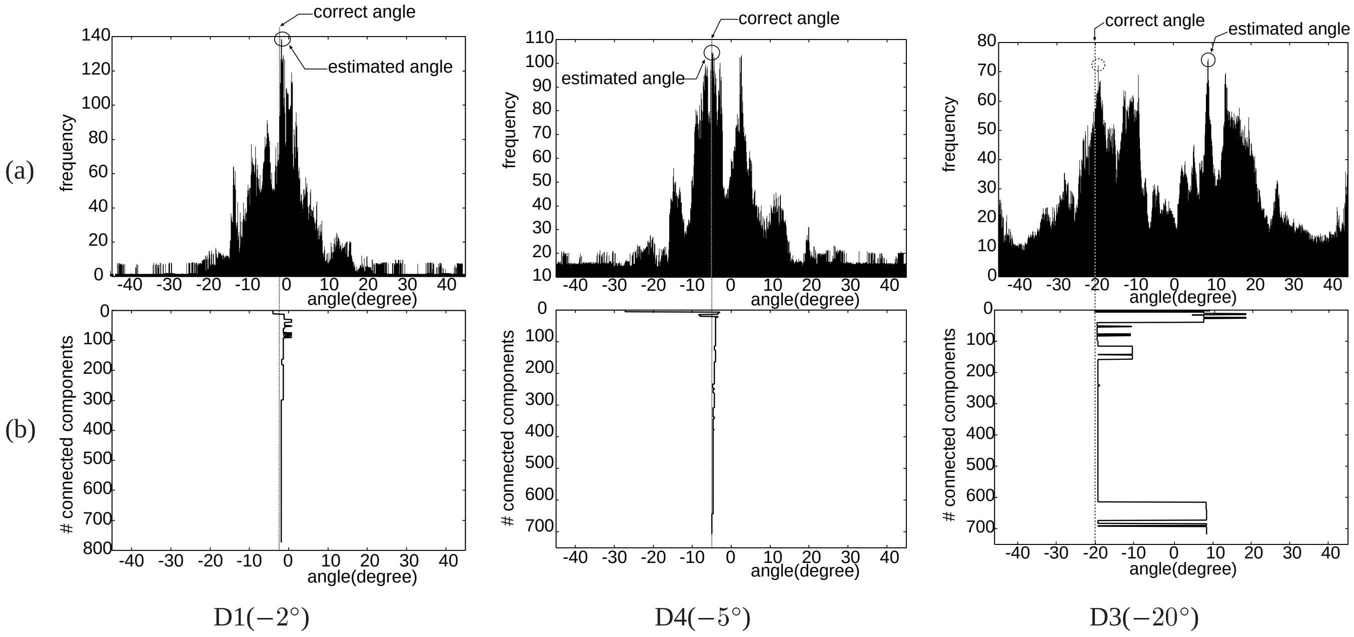


Figure 8. Estimation result. (a) Voting result and estimated skew angle as the peak of the voting result. (b) Change of estimation result according to the number of characters (∼ connected components).

Table 1. Statistics of absolute errors of estimated skew angles.

$\leq 0.5^\circ$	$\leq 1.0^\circ$	$\leq 1.5^\circ$	$\leq 2.0^\circ$
20/55 (36%)	45/55 (82%)	49/55 (89%)	54/55 (98%)

Table 2. Absolute estimation error (degree) at each document image.

skew (°)	D1	D2	D3	D4	D5	average
-30	0.7	0.7	0.7	0.7	0.4	0.6
-20	0.7	0.7	29	0.7	0.4	6.3
-10	0.5	0.9	0.9	0.2	0.3	0.6
-5	0.7	0.7	1.5	0.0	0.2	0.6
-2	0.1	0.9	0.2	0.2	0.2	0.3
0	1.7	1.7	1.7	1.7	0.4	1.4
2	0.6	0.6	0.4	0.3	0.2	0.4
5	0.3	0.3	0.9	0.9	0.3	0.5
10	1.0	1.4	2.0	1.4	0.2	1.2
20	0.7	0.7	0.9	0.7	0.6	0.7
30	0.7	0.7	1.0	1.2	0.4	0.8
average	0.7	0.9	3.6	0.7	0.3	1.2

4.3. Accuracy of estimated skew angles

Table 1 summarizes the absolute errors of the skew angles estimated for the 55 test document images. For 98% (=54/55) of the test images, the absolute error was less than 2.0°. Table 2 shows the absolute error for each of 55 test images. This table indicates that the estimation accuracy does not depend on the skew angles. The table also indicates that the estimation accuracy is not degraded by the existence of mathematical expressions, that is, the undefined categories. The skew of only one test image ("D3 rotated −20°") was poorly estimated. The reason of this failure will be discussed later.

It is noteworthy that the skew angles of D5 were also estimated successfully. This success emphasizes the usefulness of the proposed method since the conventional methods assuming straight text lines will fail to estimate the skew angles of D5.

Fig. 8 (a) shows the histogram of the skew angle candidates, i.e., the voting result. The first and the second histograms have their peaks at the correct skew angles. Consequently, the correct skew angles were estimated. The third histogram is of the failure result ("D3 rotated −20 °"); the histogram has two rivaling peaks and the false peak won by a slight difference.

Fig. 8 (b) shows the change of the peak according to the increase of connected components. The first and the second

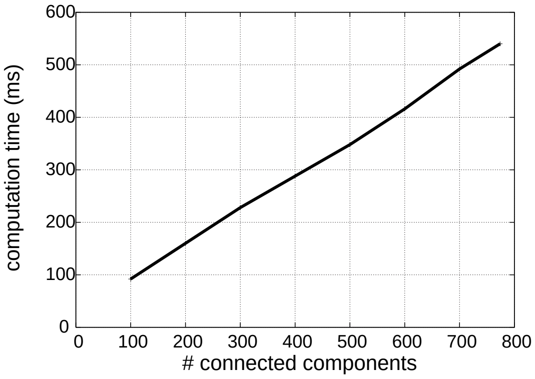


Figure 9. Computation time.

results, i.e., successful results, show quick convergence to the correct peak by $100 \sim 300$ connected components. The third result, i.e., the failure result, also shows the convergence to the correct peak by $200 \sim 600$ connected components; the peak, however, has moved to false one by 600 or more connected components.

4.4. Failure analysis

One of the main reasons of having the false peak in "D3 rotated $-20°$" was the error of the invariant. The invariants of several categories were very sensitive to the noise at the image acquisition. Especially, the invariant sometimes shows a drastic difference from the stored one by the disappearance of thin "serifs" by the low resolution. (As noted above, the resolution of 600 dpi was used on preparing the test images, whereas 1440 dpi was used on preparing the instances.) Since a false invariant always leads to false category estimation, it consequently leads to absolutely false skew estimation.

4.5. Computation time

Fig. 9 shows the computation time (CPU: Intel Pentium D) as a function of the number of connected components used for estimating the skew angle. From this graph, it is shown that the proposed method requires 100∼ 200 ms for each document; this is because the proposed method could reach the correct skew angle with about 100∼300 connected components (Fig. 8(b)).

5. Estimation of other deformations

Although we have focused on the skew estimation problem in this paper, the proposed method can estimate other geometric deformations by using different invariant and variant. For example, it is possible to estimate the shear deformation of objects which undergo affine transformation. In this case, for estimating the shear η, we will use an affine invariant q_c for estimating the category and an affine variant $p_c(\eta)$ which is a similarity invariant but a shear variant.

The estimation of perspective deformation, which is the most important deformation for camera-based OCR, can also be tackled by the proposed method. This can be realized by the fact that the perspective deformation can be decomposed into affine transformation (6 degrees of freedom) and the perspective component that controls the line at infinity (2 degrees of freedom) [7]. Let ϕ and ψ denote two parameters specifying the perspective component. In this case, we can estimate and compensate the perspective deformation according to the following steps:

1. Estimate the category c of each connected component $\boldsymbol{X}$ by using a perspective invariant q_c.

2. Estimate ϕ and ψ by using a perspective variant $p_c(\phi, \psi)$ which is an affine invariant and a variant to ϕ and ψ. The voting for this estimation is performed on the two-dimensional (ϕ, ψ)-plane.

3. Compensate the perspective component by using the estimated ϕ and ψ. The resulting document image will only undergo an affine transformation.

4. Estimate and compensate the shear η by the procedure described at the beginning of this section.

5. Finally, estimate and compensate the rotation θ by the procedure of Section 3.

Note that if we use another variant $p_c(\eta, \theta)$, the last two steps can be unified into one step.

6. Conclusion and Future Work

A novel technique for estimating document skew (i.e., rotation) has been proposed and its performance has been evaluated quantitatively and qualitatively via several experiments. The proposed method estimates the skew angle of each connected component very efficiently by using a rotation invariant and a rotation variant. The skew angles estimated at individual connected components are subjected to a voting procedure to find the most reliable skew angle as the entire document skew. The experimental result on 55 document images has shown that the skew angles of 54 document images were successfully estimated with errors smaller than 2.0°. The computational feasibility was also certified experimentally.

Future work will focus on the following points:

- Improvement of invariant. As analyzed in 4.4, the proposed method fails mainly due to the error of the invariant. Erroneous invariants always lead to false category candidates and thus lead to absolutely false skew estimation. A more stable and distinguishable invariant will be necessary. A combinatorial use of multiple invariants is a possible remedy.

- Removal of less reliable instances. In this paper, the instance of the variant was prepared for every categories. Several instances, however, were less reliable as pointed in Section 3.2.3; for example, the variants of “o”-shaped characters do not change drastically by rotation and thus not express the skew angle clearly. In addition, the invariants of several categories are sensitive to noise. Removal of those invariants and variants will exclude false category candidates and skew angle candidates.

- Estimation of deformations other than rotation. As noted in Section 5, the proposed method can be extended to estimate various deformations by using suitable combinations of variants and invariants. Especially, perspective deformation will be the most important one for camera-based OCR.

References

[1] J. Liang, D. Doermann and H. Li: “Camera-based analysis of text and documents: a survey,” Int. J. Doc. Anal. Recog., vol. 7, pp. 84–104, 2005.

[2] U. Pal, M. Mitra, B. B. Chaudhuri, “Multi-skew detection of Indian script documents,” Proc. Int. Conf. Doc. Anal. Recog., pp. 292–296, 2001.

[3] Y. Ishitani, “Document Skew Detection Based on Local Region Complexity,” Proc. Int. Conf. Doc. Anal. Recog., pp. 49–52, 1993.

[4] X. Jiang, H. Bunke, and D. Widmer-Kljajo, “Skew Detection of Document Images by Focused Nearest-Neighbor Clustering,” Proc. Int. Conf. Doc. Anal. Recog., pp. 629–632, 1999.

[5] Y. Lu and C. L. Tan, “Improved Nearest Neighbor Based Approach to Accurate Document Skew Estimation,” Proc. Int. Conf. Doc. Anal. Recog., pp. 503–507, 2003.

[6] S. Lu and C. L. Tan, “Camera Document Restoration for OCR,” Proc. Int. Workshop Camera-Based Doc. Anal. Recog., pp. 17–24, 2005.

[7] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2nd edition, 2004.

Poster Papers

A Model-based Book Dewarping Method Using Text Line Detection

Bin Fu, Minghui Wu, Rongfeng Li, Wenxin Li, Zhuoqun Xu, Chunxu Yang

State Key Laboratory on Machine Perception
School of Electronics Engineering and Computer Science
Peking University
Beijing, China
{ fubinpku, wuminghui, rongfeng, lwx, zqxu }@pku.edu.cn

Abstract

In this paper, we propose a book dewarping model based on the assumption that the book surface is warped as a cylinder. This model extends the model proposed by Cao and makes Cao's model a special case of our model. This extension removes the constraint of Cao's model that the camera lens must be strictly parallel to the book surface, which is hard to make in practice, therefore enables a user to take a picture from different point of views conveniently. The main idea of the model is to build up the correspondence between a rectangle region on a flat surface and its curved region on the distorted book image and the dewarping task is to flatten the curved region to its original rectangle shape. The experimental results demonstrate the effectiveness of our proposed book dewarping approach.

1. Introduction

A number of camera document image restoration techniques have been reported in the literature. According to whether auxiliary hardware is required, the proposed techniques can be classified into two different categories. Approaches [1-2] with auxiliary hardware can solve the dewarping problem well; however, the costly equipment makes the "hard" approaches unattractive. Approaches without auxiliary hardware can be further classified into two classes according to the problem it is oriented. Approaches [3-4] focus on the problem of removing the perspective distortion in images of flat documents. Approaches [5 - 10] focus on a more complex situation: Page warping adds a non-linear, non-parametric distortion on the perspective document image. In [5], Ulges propose a line-by-line dewarping approach and In [6], Lu uses a document image segmentation algorithm. Both the two methods are sensitive to the resolution and language in a document image. In [7], Cao models the book surface as a cylinder, and requires the camera lens parallel to the generatrix of the book surface. This constraint makes the approach inconvenient in practice. In [8], Liang makes use of the concept of developable surface to model the warped surface, and use text information to estimate the surface, although this approach can handle complex warping situation, such as page curling at the corner, it is time consuming and needs to get the text lines precisely. In [9], Ezaki uses a group of cubic splines to model the warped text lines and proposes a global optimization algorithm to find the proper cubic splines, their approach is novel and can lessen the curl degree, but they did not show satisfied rectification results in their paper.

In this paper, we propose a transform model to stretch a cylinder surface into a flat one. This model is further applied to solve the problem of restoring document images, which suffer from perspective and page curling distortion. Comparing with Cao's approach [7], our model does not require the camera lens to be parallel to the book page surface's generatrix; therefore it imposes fewer restrictions on the input image, so it is more widely applicable.

The rest of this paper is organized into four sections. Section 2 introduces our dewarping model. Section 3 focuses on the rectification process. We present our experimental results in Section 4, and Section 5 concludes the paper.

2. The Transform Model

Assume we have a rectanglar area on a page surface as shown in Figure 1. The existence of bookbinding often causes the distortion of the book image. The projection of the book image is shown in Figure 2. Figure 3 is the distorted image of the book with its rectangle area.

Before the transformation, we make the following assumptions:

(1) The borders of the rectangle are paralleled to the borders of the page.

(2) The page surface is warped as a cylinder.

(3) A cylinder is a surface generated by a straight line intersecting and moving along a closed planar curve. The straight line is called the generatrix, and the closed planar curve is called the directrix.

(4)The left and right borders of the page are parallel to the generatrix of the cylinder.

(5) Because the left and right borders of the rectangle are parallel to the left and right borders of the page, they are therefore parallel to the generatrix. In this case, the images of the left and right borders of the rectangle are straight line segments.

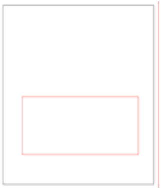


Figure 1. A rectangle area on a page surface

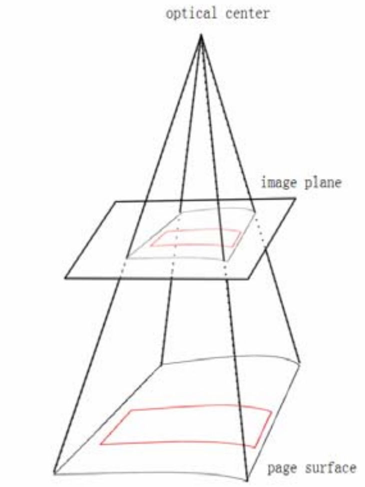


Figure2. The projection of the page surface

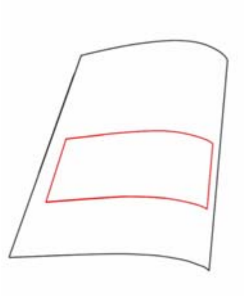


Figure 3. The distorted image of the page surface with its rectangle area

Our goal is to generate a transformation to flatten the image in Figure 3 to its original image in Figure 1. The transformation is a mapping from the curved coordinate system to a Cartesian coordinate system. Once curved coordinate net is set up on the distortion image in Figure 4, the transformation can be done in two steps: First, the curve net is stretched to a straight one (Figure 5) and then adjusted to a well-proportioned square net (Figure 6).

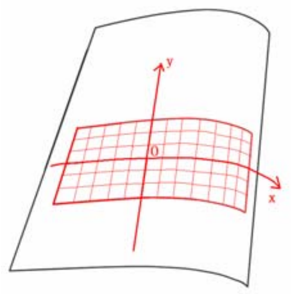


Figure 4. An example of a warp document page with its curved coordinate net

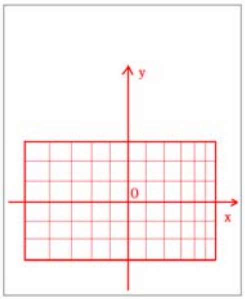


Figure 5. The square net generated by stretching the curved coordinate net in Figure 4.

2.1 Stretched the Curve Coordinate Net

In this step, our goal is to transform a picture like Figure 4 into a picture like Figure 5. We may also view the transform process from the other direction. That is, given the matrix on Figure 5, calculate each

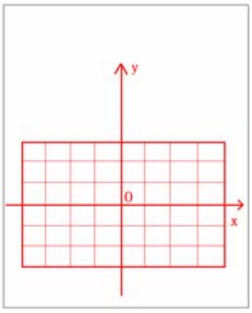


Figure 6. The final result square net generated by adjusting the width in Figure 5 according to the curvature of the page surface.

cell's correspondence point on Figure 4 and fill each cell with the content of its correspondence point. We fill the matrix of Figure 5 in two steps: 1) let the page lower boundary in Figure 4 map with the bottom line in Figure 5; 2) one by one line up the page lower boundary point in Figure 4 and the vanish point (transform center). The line intersects the page upper boundary in Figure 4 at a point and then we get a line segment on Figure 4. Map this line segment with the correspondence column in Figure 5. In step 2, we cannot evenly distribute the points on the line segment in Figure 4 to match with the column in Figure 5 due to their different distances to the lens. In the following section, we explain how to match the line segment in Figure 4 to a column in Figure 5.

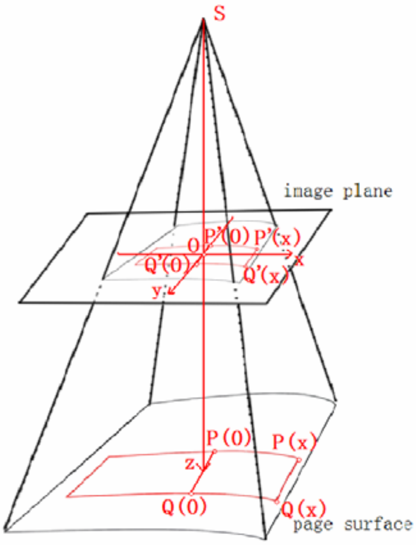


Figure 7. The details of the projection between an arbitrary line segment P'(x)Q'(x) which across the transform center and its inversimage P(x)Q(x)

In Figure 7. P'(x)Q'(x) is a line segment (its x coordinate is x) which along the y coordinate in the image plane and P(x)Q(x) is its inverse image on the page surface. S is the lens. O is the intersection of the optical axis and the image plane. Here the image plane may not be parallel to the page surface due to the variation of direction of the lens. Figure 8 shows the yOz plane of Figure 7. The z axis meets the page surface at R(x). I(x) and H(x) are the foot points of Q(x) and P(x) on the z axis.

By geometry, we get the following equations:

$$\frac{SO}{SH(x)} = \frac{P'(x)O}{P(x)H(x)}, \quad \frac{SO}{SI(x)} = \frac{Q'(x)O}{Q(x)I(x)},$$

$$P'(x)O = \frac{P(x)H(x)\cdot SO}{SH(x)} = \frac{P(x)R(x)\sin\angle P(x)R(x)H(x)\cdot SO}{SR(x)+R(x)H(x)} \qquad (1)$$

$$= \frac{SO\cdot P(x)R(x)\sin\angle P(x)R(x)H(x)}{SR(x)+\cos\angle P(x)R(x)H(x)\cdot P(x)R(x)}$$

Let $\alpha = \angle P(x)R(x)H(x), f = SO, d(x) = SR(x),$, we get

$y'_P(x) = P'(x)O, y_P(x) = P(x)R(x)$

$$y'_P(x) = \frac{fy_P(x)\sin\alpha}{d(x)+y_P(x)\cos\alpha} \qquad (2)$$

Let $a = \dfrac{d(x)}{f\sin\alpha}, b = \dfrac{\cos\alpha}{f\sin\alpha}$,

$y_P = y_P(x), y'_P(x) = y'_P$, then

$$y'_P = \frac{y_P}{a+by_P} \qquad (3)$$

Analogously, let $y'_Q = Q'(x)O, y_Q = Q(x)R(x)$, we have the equation:

$$y'_Q = \frac{y_Q}{a+by_Q} \qquad (4)$$

Now we consider $P_0, P_1, \ldots, P_n$ as a series of equidistant points on P(x)Q(x), which have the y coordinates: $y_i = \dfrac{n-i}{n}y_P + \dfrac{i}{n}y_Q$, we can get the y coordinates of the image points $P_0', P_1', \ldots, P_n'$:

$$y'_i = \frac{y_i}{a+by_i} \qquad (5)$$

y'_P, y'_Q are the y coordinates of the points P(x) and Q(x) which are on the edge of the page. They can be detected by page extraction in the next section. In practice, we let $y_P = y_Q = k$, where k is a constant value obtained by experience. Thus, with the knowledge of y'_P, y'_Q and y_P, y_Q, we can calculate the parameters *a* and *b* with the equations (3) and (4).

Using (5), the line segment P'Q' is divided into n pieces. Dividing all the line segment along the y coordinate, the coordinate net in Figure2 is set up. Thus, the coordinate net in Figure3 is generated by a one-to-one mapping.

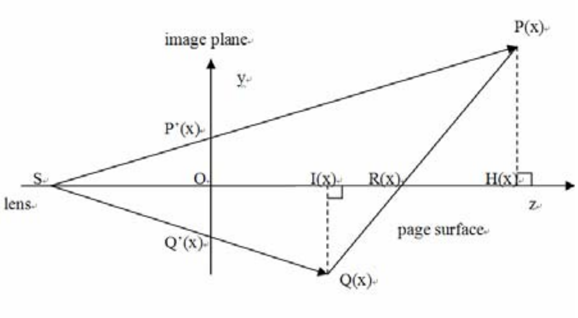


Figure 8. The yOz plane of figure 5

2.2 Adjust the Width of the Square Net

Consider the surface of the page as a cylinder whose curve equation is *b(x)* (Figure 9). In order to adjust the *x* coordinate to become well-proportioned, for each point we adjust its x coordinate x_{adjust} into its curve distance *j(x)* from the book spine.

$$x_{adjust} = j(x) = \int_0^x \sqrt{(db(x))^2 + (dx)^2} = \int_0^x \sqrt{1 + b'(x)^2}\,dx \qquad (6)$$

In Figure 7, P(0)Q(0) is the book spine, their x coordinate is 0. P'(0), Q'(0) is the image of P(0),Q(0). We can get b(x) by the y coordinate of P'(x),P'(0) (that is $y'_P(x), y'_P(0)$). Figure 10 shows the yOz plane of Figure 7.

In Figure 10,

$$SR(x) = SR(0) - R(0)R(x) = SR(0) - \frac{b(x)}{\sin \angle P(0)R(0)R(x)} \qquad (7)$$

Thus,

$$d(x) = d(0) - \frac{b(x)}{\sin\alpha} \qquad (8)$$

In equation (2), ignoring the variety of $y_P(x)$, let $y_P = y_P(x)$,

$$y'_P(x) = \frac{f y_P \sin\alpha}{d(0) - \frac{b(x)}{\sin\alpha} + y_P \cos\alpha} \qquad (9)$$

$$y'_P(x) - y'_P(0) = \frac{f y_P \sin\alpha}{d(0) + y_P\cos\alpha - \frac{b(x)}{\sin\alpha}} - \frac{f y_P \sin\alpha}{d(0) + y_P \cos\alpha} \qquad (10)$$

$$= \frac{f y_P b(x)}{(d(0) + y_P\cos\alpha)(d(0) + y_P\cos\alpha - \frac{b(x)}{\sin\alpha})}$$

Assuming that b(x) << $d(0) + y_P \cos\alpha$,

$$y'_P(x) - y'_P(0) = \frac{f y_P b(x)}{(d(0) + y_P \cos\alpha)^2} = Cb(x) \qquad (11)$$

where *C* is a constant and in practice we used an experience number for it.

Using (11) we get b(x). Assuming b'(x) is the derivative of b(x), the square net can be adjusted to be a well-proportioned one (Figure 6) with the following equation:

$$x_{adjust} = \int_0^x \sqrt{1 + b'(x)^2}\,dt \qquad (12)$$

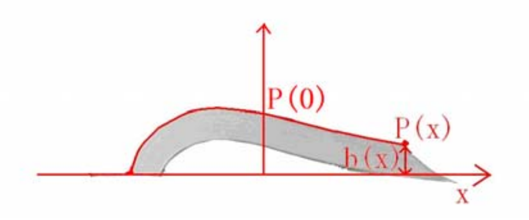


Figure 9. The curve b(x) of the page surface

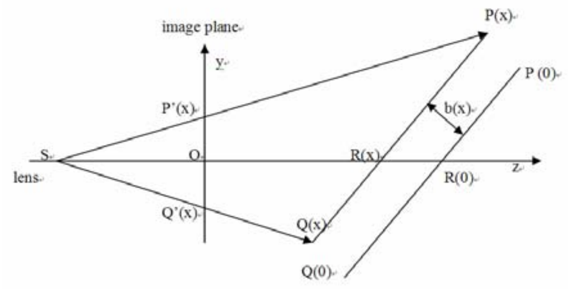
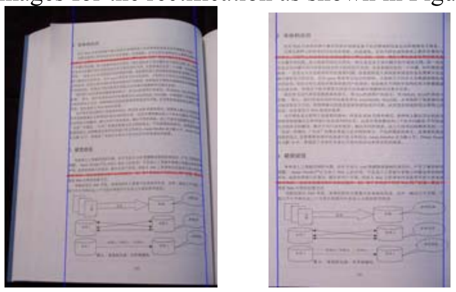


Figure 10. The yOz plane of Figure 5.

3. The Proposed Approach

According to the transform model described in Section 2, we need two line segments and two curves to dewarp a cylinder image, so our task is to find the left and right boundaries and top and bottom curves in book images for the rectification as shown in Figure 11:



(a) (b)

Figure 11. The illustration of document image before and after rectification (a) Original book image; (b) Dewarped book image.

The rectification process involves three steps: 1) the text line detection, 2) left and right boundary estimation and top and bottom curves extraction, and 3) document rectification. The flowchart of our rectification process is illustrated in Figure 12:

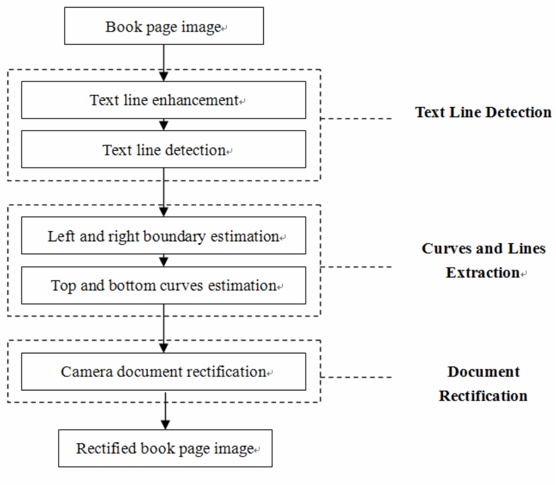
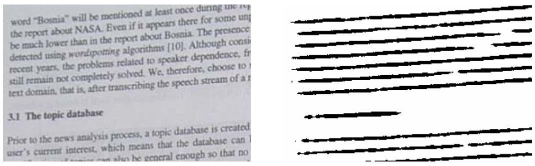

Book page image
Text line enhancement
Text line detection
Text Line Detection
Left and right boundary estimation
Top and bottom curves estimation
Curves and Lines Extraction
Camera document rectification
Document Rectification
Rectified book page image

Figure 12. The flowchart of the proposed approach.

3.1 Text Line Detection

The text line detection process includes two steps. The first step makes characters on one text line connected to each other and the second step classifies the pixels on different text lines into different collection for further curve fitting.

3.1.1 Text line enhancement. S This step uses a method similar to the method proposed by Ching-Tang Hsieh in [10]. Figure 13 shows the character connection results.



word "Bosnia" will be mentioned at least once during the
the report about NASA. Even if it appears there for some un
be much lower than in the report about Bosnia. The presence
detected using *wordspotting* algorithms [10]. Although consi
recent years, the problems related to speaker dependence, fr
still remain not completely solved. We, therefore, choose to
text domain, that is, after transcribing the speech stream of a

3.1 The topic database

Prior to the news analysis process, a topic database is created
user's current interest, which means that the database can
also be general enough so that no

(a) (b)

Figure 13. Character connection result (a) Original book image (b) Result image.

3.1.2 Pixel classification. In this step, we first detect the mid points of all text lines, and then with each mid point, we trace all the pixels belong to the same text line as it. Algorithm 1 gives the details of the pixel classification process.

Algorithm 1: Text line point collection with automatic line direction adjustment:

Input: Thinned image: $T_{m \times n}$ ' , Text line mid point set: $\mathrm{Set}_{\mathrm{midpoint}}$

Output: Point set of each text line SetP_i

For each Point (x_i, y_i) in $\mathrm{Set}_{\mathrm{midpoint}}$, the algorithm detects points in both left and right directions.

Step 1: detect points on the left of Point(x_i, y_i) at the same text line, as is illustrated in Figure 14. We use a step of L pixels and a detection height of H pixels (In our experiment, L = 10, while H = 6), the detection skew is set to K_t.

Let (x_{t-1}, y_{t-1}) be the point detected in the previous step, and (x_{t-2}, y_{t-2}) be the point detected before (x_{t-1}, y_{t-1}).

The skew of line direction can be calculated using the following formula (K_0 is set to be 0) :

$$K_t = K_{t-1} + (y_{t-1} - y_{t-2}) / (x_{t-1} - x_{t-2}) / 10 \qquad (15)$$

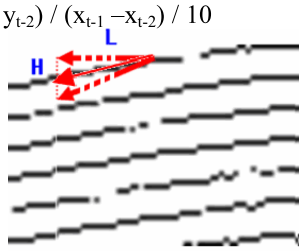
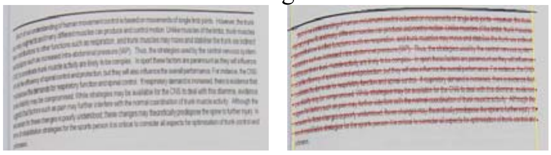


Figure 14. Text line collection step.

Step 2: a similar trace process as step 1 is used to collect all points on the right side of the mid point and the direction is to the right of the mid point of text line.

Step3: combine point sets got in step1 and step2 into SetP_i, this point set is then the point set of text line_i.

After Text line collection, all points belong to a same text line are collected. Using a polynomial curve fitting algorithm, we can get a set of polynomial curves which represent the text lines. Figure 15 illustrates the result of text line detection algorithm.



(a) (b)

Figure 15. Example of (a) Part of original book image, (b) Text line curve extracted using our proposed method - the curves across the characters are extracted text lines.

3.2 Feather curves and lines extracting

Once we get all the text lines, we estimate the left and right boundaries of the text lines and choose two text lines as the basis for transformation.

3.2.1 Left and right boundary estimation. The algorithm for estimating the left and right boundaries can be formalized as follows:

Algorithm 2: left and right boundary estimation

Input: End point set (x_i, y_i) of text lines

Output: Boundaries of columns.

Step 1: For all the end points on the left sides of all text lines, least square estimate (LSE) method is used to get a straight line L.

Step 2: Calculate the distance between (x_i,y_i) and L as D(i). Eliminate (x_i,y_i) if $D(i) > T_a$ and (x_i,y_i) is at the right of L. (T_a is a threshold for eliminate the end points of text lines which are indent text lines or short text lines)

Step 3: do step 1 and step 2, until each $D(i) < T_b$. (T_b is a threshold to make sure L be close to each end point of text lines)

Then L is considered the left boundary of this page.

A similarly iterative process is used for right boundary estimation.

The result of boundary estimation is illustrated in Figure 16:

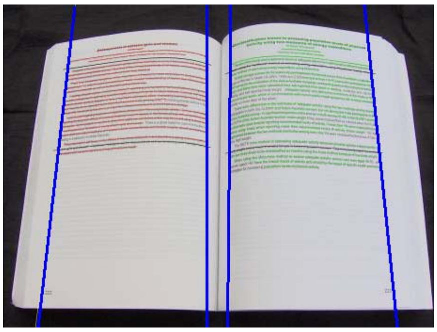


Figure 16. Left and right boundaries.

3.2.2 Top and bottom curves selection algorithm. The algorithm for selecting top and bottom curves can be formalized as follows:

Algorithm 3: Top and bottom curve selection algorithm

Let D_{li} represents the distance between the left end point of text line i and the left boundary of text lines. D_{ri} represents the distance between the right end point of text line i and the right boundary. N_{line} be the amount of text lines.

Top text line is the line with the smallest i which following the formula $(D_l + D_r) * i / N_{line} < Td$, and bottom text line is the line with the largest i which following the formula $(D_l + D_r) * (N_{line} - i) / N_{line} < Td$.

These formulas guarantee that the two lines selected are not too close to each other, and their end points are not too far from the left and right boundaries.

3.3 Document Rectification Algorithm

After estimating the left and right boundary and top and bottom curves, Rectification algorithm is applied to dewarp book image. Let curve1 denotes the curve segment generated by the top curve cut by the left and right boundaries, and curve2 denotes the bottom curve cut by the left and right boundaries.

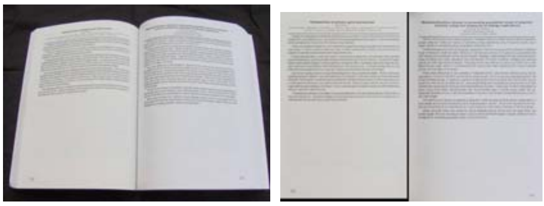
Algorithm 4: Document Rectification

Input: Original Image – $I_{m\times n}$, Curve1, Curve2

Output: Output Image - $O_{m\times n}$

```
Step 1: Calculate b(x) from Curve1,Curve2 according to equation (11)
Step 2: For each point P on Curve1, Q on Curve2, PQ is paralleled to the directrix.
    1. Calculate a, b according to equation (3) and (4).
    2. For each point on line segment PQ,
        a. Calculate its new y coordinate with (5) and x coordinate with (12)
        b. Get its color from $I_{m\times n}$ and set it to $O_{m\times n}$ on
           its new coordinate
```

After algorithm 4, the original image is rectified. The rectification result is illustrated in Figure 17.



(a) (b)

Figure 17. The transform result of a book image. (a) is the original image, (b) is the transformed result.

4. Experimental results and analysis

We implemented the algorithms in C# and run it on a personal computer equipped with Windows XP, and AMD 3600+ AM2 CPU, 1G Memory. An image database of 100 distorted book images that contain different languages is used for the testing. We captured the 100 book images according to the following input requirements:

1) The distance between the camera lens and page surface is about 50 - 100cm.

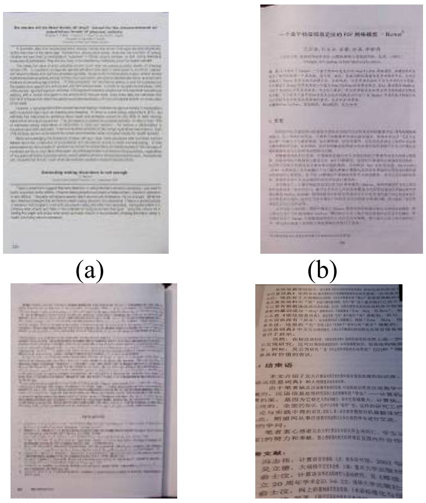
2) The angle between the camera lens and the book surface is less than 30 degree.

3) Book image should contain enough long text lines.

4) Book image need not contain the whole page but the text area of interest.

A Kodak DX7630 digital camera with a capture resolution of 2856 x 2142 - 6 Mega pixel is used, and images are taken under day light with dark background. Six books with different contents are involved, one book is in Chinese, four are books in English, and one is bilingual. Among the 100 images, ten percent contain pictures.

We evaluate the performance of the proposed method based on whether the text lines of a book image are straight or not and the sizes of characters on the same text line are uniform. Results are labeled as “A+”, “A”, “B”, or “C”. “A+” means the text lines in the dewarped result do not have any wave and looks exactly like generated by a scanner. “A” means the text lines have little waves, but still comfortable for reading. “B” means the dewarping result have waves and affect the feeling. “C” means the dewarping result has large waves or is hard for reading. Figure 18 illustrates the classifications.



(a) (b)

(c) (d)

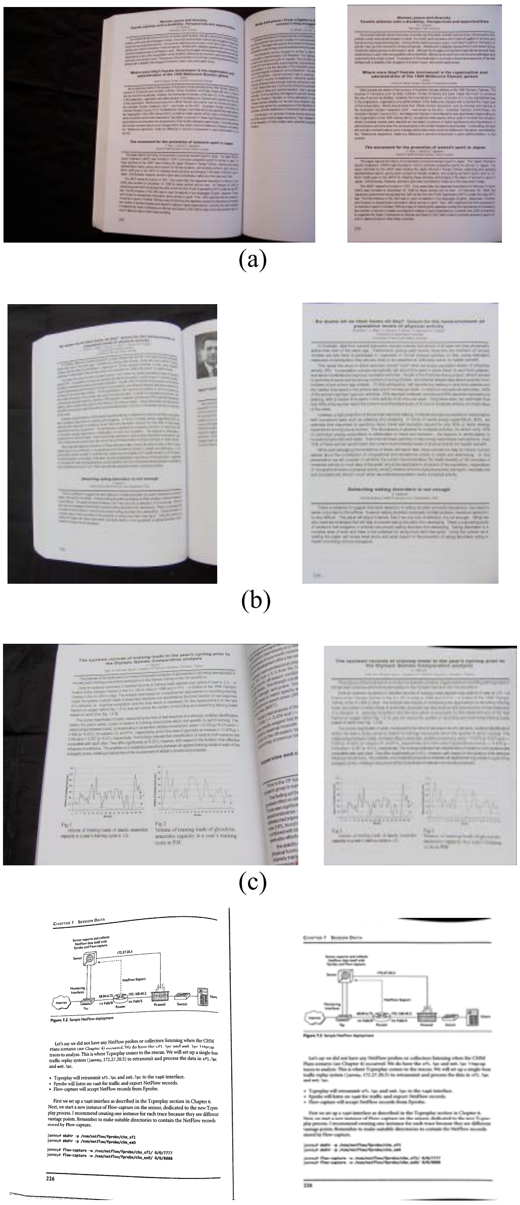
Figure 18. Sample images from each category (a) “A+” (b) “A” (c) “B” (d) “C”.

The experimental results are illustrated in Table 1.

Table 1. Test result

Classification	Percentage
A+	55%
A	35%
B	9%
C	1%

Some results are illustrated in Figure 19. Figure 20 provides a closer view of 19(a):



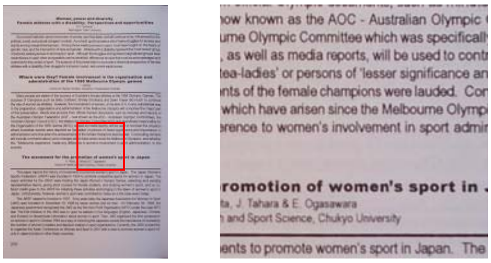
(a)

(b)

(c)

(d)

Figure 19. Transformation results.



now known as the AOC - Australian Olympic
urne Olympic Committee which was specificall
, as well as media reports, will be used to contr
tea-ladies' or persons of 'lesser significance an
nts of the female champions were lauded. Cor
which have arisen since the Melbourne Olymp
erence to women's involvement in sport admi
romotion of women's sport in .
ta, J. Tahara & E. Ogasawara
n and Sport Science, Chukyo University
ents to promote women's sport in Japan. The

(a) (b)

Figure 20. (a) is Figure 21(a) with a rectangle on it; (b) is a close view of (a) .

We also use the OCR software Tsinghua TH-OCR 9.0 to make a comparison of OCR qualities between before and after rectification of 20 book images. 10 books are in English and 10 books in Chinese. There are totally 9212 English words on English books. And 7102 Chinese characters on Chinese books. 5923 English words are correctly recognized before rectification, while 8788 words recognized after rectification. 305 Chinese characters are correctly recognized before rectification, while 6726 characters recognized after rectification.

Table 2 illustrates the OCR rates before and after rectification.

Table 2. OCR Test result

Book Language	Before Rectification (OCR rate)	After Rectification (OCRrate)
English	64.3%	95.4%
Chinese	4.3%	94.7%

Our experimental results reveal that the coordinate transform model and document rectification approach proposed in this paper can rectify both perspective distortion and warping well. It ensures effective layout analysis and word segmentation, hence bringing about higher reading performance and less recognition errors.

5. Conclusion

In this paper, we propose a model for dewarping a cylinder image into a flat one. This model is applied to the problem of rectifying document image taken by a digital camera, which suffers from perspective distortion and document surface warping. Our experimental results show that our approach can dewarp the curved surface well, and it is efficient on warped document images of different languages.

Further improvement in the processing speed and de-blurring of the shade area will be explored. As digital camera is developed rapidly, the proposed document restoration technique may open a new channel for document capture and digitalization.

Acknowledgements

This work is financial supported by Laserfiche. The authors would also like to thank and Kurt.Rapelje, Ryan.Wong and Ed Heaney from Laserfiche for their valuable comments on this work.

Reference

[1] M. S. Brown and W. B. Seales, “Image restoration of arbitrarily warped documents”. IEEE Transactions on Pattern Analysis and Machine Intellegence, 26(10):1295–1306, October 2004.

[2] A.Yamashita, A. Kawarago, T. Kaneko, K. T. Miura, “Shape Reconstruction and Image Restoration for Non-Flat Surfaces of Documents with a Stereo Vision System”, International Conference on Pattern Recognition, vol. 1, August 2004, Cambridge, UK, page 482 - 485.

[3] M. Pilu, “Extraction of illusory linear clues in perspectively skewed documents”. In Proceedings of the Conference on Computer Vision and Pattern Recognition,
volume 1, pages 363–368, 2001.

[4] P. Clark, M. Mirmhedi, “Rectifying perspective views of text in 3Dscenes using vanishing points”, Pattern Recognition, vol. 36, pp. 2673-2686, 2003

[5] A. Ulges, C. H. Lampert, and T. M. Breuel, “Document image dewarping using robust estimation of curled text lines”, in Proceedings of the Internatioanl Conference on Document Analysis and Recognition, 2005, pp. 1001–1005

[6] Shijian Lu, Chew Lim Tan, “The Restoration of Camera Documents through Image Segmentation”, Analysis Systems, 2006, pp. 484-495

[7] H. Cao, X. Ding, and C. Liu, “Rectifying the bound document image captured by the camera: A model based approach”. In Seventh International Conference on Document Analysis and
Recognition - ICDAR2003, pages 71–75, 2003.

[8] Jian Liang, Daniel DeMenthon, David Deormann “Flattening Curved Documents in Images”, International Conference on Computer Vision and Pattern Recognition, 2005, pp. 338-345.

[9] Hironori Ezaki and Seiichi Uchida, “Dewarping of document image by global optimization”, in Proc. 8th International Conference on Document Analysis and Recognition, 2005, pp. 302-306.

[10] Ching-Tang Hsieh, Eugene Lai, You-Chuang Wang “An effective algorithm for fingerprint image enhancement based on wavelet transform” Pattern Recognition 36 (2003) 303 – 312”.

Automatic Borders Detection of Camera Document Images

N. Stamatopoulos, B. Gatos, A. Kesidis
Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,
National Center for Scientific Research "Demokritos", GR-153 10 Athens, Greece
http://www.iit.demokritos.gr/cil/,
{nstam, bgat, akesidis}@iit.demokritos.gr

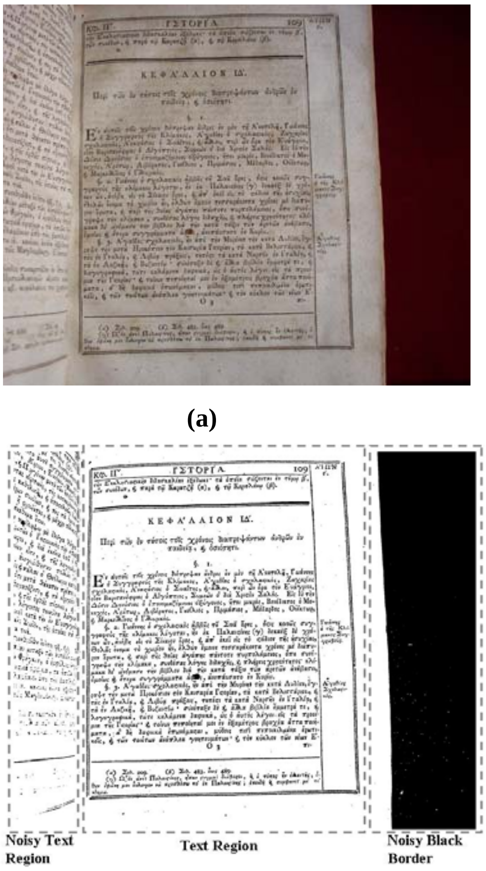
Abstract

When capturing a document using a digital camera, the resulting document image is often framed by a noisy black border or includes noisy text regions from neighbouring pages. In this paper, we present a novel technique for enhancing the document images captured by a digital camera by automatically detecting the document borders and cutting out noisy black borders as well as noisy text regions appearing from neighbouring pages. Our methodology is based on projection profiles combined with a connected component labelling process. Signal cross-correlation is also used in order to verify the detected noisy text areas. Experimental results on several camera document images, mainly historical, documents indicate the effectiveness of the proposed technique.

1. Introduction

Document images are often framed by a noisy black border or include noisy text regions from neighbouring pages when captured by a digital camera. Approaches proposed for document segmentation and character recognition usually consider ideal images without noise. However, there are many factors that may generate imperfect document images. When a page of a book is captured by a camera, text from an adjacent page may also be captured into the current page image. These unwanted regions are called "noisy text regions". Additionally, there will usually be black borders in the image. These unwanted regions are called "noisy black borders". Figure 1 shows an example of an image having noisy black borders as well as noisy text regions. All these problems influence the performance of segmentation and recognition processes.

There are only few techniques in the bibliography for page borders detection. Most of them detect only noisy black borders and not noisy text regions. Fan et al. [1] proposes a scheme to remove the black borders of scanned documents by reducing the resolution of the document image and by marginal noise detection and removal. Le and Thoma [2] propose a method for border removal which is based on classification of blank/textual/non-textual rows and columns, location of border objects, and an analysis of projection profiles



(a)

(b)

Figure 1. Example of an image with noisy black border, noisy text region and text region. (a) Original camera document image (b) Binary document image

and crossing counts of textual squares. Avila and Lins [3] propose the invading and non-invading border algorithms which work as “flood–fill” algorithms. The invading algorithm, in contrast with non-invading, assumes that the noisy black border does not invade the black areas of the document. Finally, Avila and Lins [4] propose an algorithm based on “flood-fill” component labelling and region adjacency graphs for removing noisy black borders.

In this paper, a new and efficient algorithm for detecting and removing noisy black borders as well as noisy text regions is presented. This algorithm uses projection profiles and a connected component labelling process to detect page borders. Additionally, signal cross-correlation is used in order to verify the detected noisy text areas. The experimental results on several camera document images, mainly historical, documents indicate the effectiveness of the proposed technique. The rest of this paper is organized as follows. In section 2 the proposed technique is presented while experimental results are discussed in Section 3. Finally, conclusions are drawn in Section 4.

2. Proposed method

Before the noisy borders detection and removal takes place, we first proceed to image binarization using the efficient technique proposed in [5]. This technique does not require any parameter tuning by the user and can deal with degradations which occur due to shadows, non-uniform illumination, low contrast, large signal-dependent noise, smear and strain.

2.1. Noisy black border detection and removal

In this stage we detect and remove noisy black borders (vertical and horizontal) of the image. The proposed algorithm which is mainly based on horizontal and vertical profiles is described in the flowchart of Fig. 2. Our aim is to calculate the limits, XB1, XB2, YB1 and YB2, of text regions as shown in Fig. 3. In order to achieve this, we first proceed to an image smoothing, then calculate the starting and ending offsets of borders and text regions and then calculate the borders limits. The final clean image without the noisy black borders is calculated by using the connected components of the image.

Consider a binary image:

$$I(x,y) = \{0,1\} \quad 0 \le x < I_x, 0 \le y < I_y \qquad (1)$$

The main modules of the proposed technique are as follows.

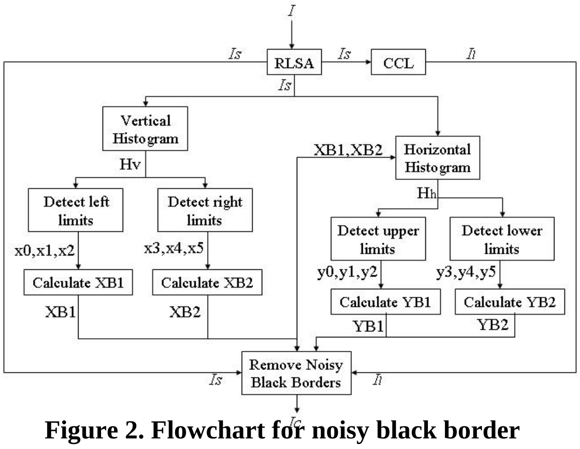


Figure 2. Flowchart for noisy black border detection and removal.

RLSA: Horizontal and vertical smoothing with the use of the Run Length Smoothing Algorithm (RLSA) [6]. RLSA examines the white runs existing in the horizontal and vertical direction. For each direction, white runs with length less than a threshold are eliminated. The empirical value of horizontal and vertical length threshold is 4 pixels. The resulting image is $I_s(x,y)$.

CCL (Connected Component Labeling): Calculate the connected components of the image $I_s(x,y)$ based on the approach described in [7]. The image consists of CS connected components C_i and the resulting labeled image is given by I_l:

$$I_l(x,y) = \begin{cases} i & \text{if } (x,y) \in C_i,\ 0 < i < CS \\ 0 & \text{othewise} \end{cases} \qquad (2)$$

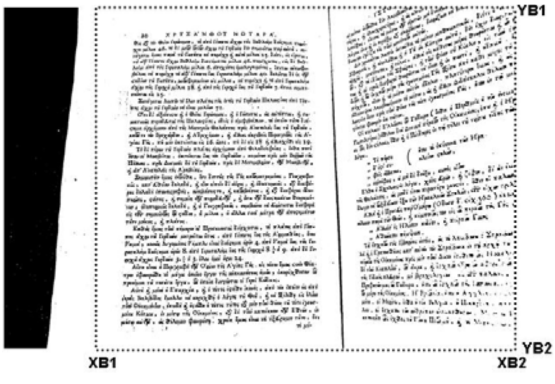


Figure 3. Limits XB1, XB2, YB1 and YB2 of text regions after noisy black border detection.

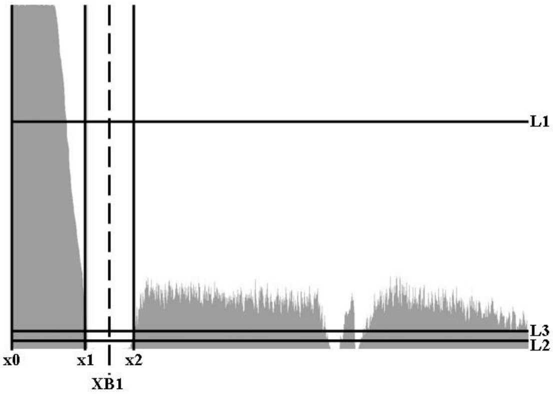


Figure 4. Projections of image in Fig. 3 and left limits detection.

Vertical Histogram: Calculate vertical histogram H_v, which is the sum of black pixels in each column.

$$H_v(x) = \sum_{y=0}^{Iy-1} I_s(x,y) \text{ where } 0 \le x < I_x \tag{3}$$

Detect left limits: Detect vertical noisy black borders in the left side of the image (see Fig. 4).

Initially we search for the start and the end (x_0, x_1) of the left vertical black border. Calculate x_0 as follows:

$$x_0 = \min(x) : (H_v(x) > L_1) \; or \; (H_v(x) < L_2) \quad \text{where } 0 \le x < I_x / 5 \tag{4}$$

The first condition, ($H_v(x) > L_1$), is satisfied when the black border starts from the left side of the image, which is the most usual case (see Fig. 3), while the second condition, ($H_v(x) < L_2$), is satisfied when white region exists before the black border. If we don't find any x_0 that satisfies the above conditions we set $x_0 = -1$, $x_1 = -1$, $x_2 = -1$ and stop this process. Otherwise, x_1 is calculated as follows:

$$x_1 = \begin{cases} \min(x) : H_v(x) < L_2 \;, x_0 < x < I_x / 2 \text{ if } H_v(x_0) > L_1 \\ \min(x) : H_v(x) > L_1 \;, x_0 < x < I_x / 2 \text{ othewise} \end{cases} \tag{5}$$

If we don't find any x_1 that satisfies the conditions we set $x_0 = -1$, $x_1 = -1$, $x_2 = -1$ and stop this process.

After we have located the black border we search for the start (x_2) of the text region (see Fig. 4) and calculate it as follows:

$$x_2 = \min(x) : \; H_v(x) < L_1 \text{ AND } H_v(x) > L_3, \; x_1 < x < I_x / 2 \tag{6}$$

If these is no x_2 satisfying Eq. (6) we set $x_2 = -1$. After experimentations, the values of L_1, L_2 and L_3 are set to: $L_1 = (2/3) * I_y$, $L_2 = (1/50) * I_y$, $L_3 = (1/20) * I_y$.

Calculate XB1: Calculate left limit (XB1) of text regions (see Fig. 3) as follows:

$$XB1 = \begin{cases} 0 & \text{if } x_0 = -1 \\ x_0 + (x_1 - x_0)/2 & \text{if } x_2 = -1 \\ x_1 + (x_2 - x_1)/2 & \text{if } x_2 \neq -1 \end{cases} \tag{7}$$

A similar process is applied in order to detect the vertical noisy black border of the right side of the image as well as the right limit XB2 of text regions.

Horizontal Histogram: Calculate horizontal histogram H_h, which is the sum of black pixels in each row at XB1 and XB2 limits.

$$H_h(y) = \sum_{x=XB1}^{XB2} I_s(x,y) \text{ where } 0 \le y < I_y \tag{8}$$

A similar process as for the vertical noisy black borders is applied in order to detect the horizontal noisy black borders as well as the upper (YB1) and bottom (YB2) limits of text regions (see Fig. 3).

Remove Noisy Black Borders: All black pixels that belong in a connected component C_i which includes at least one pixel that is out of limits are transformed in white. Finally, we get the image $I_C(x,y)$ as follows:

$$I_C(x,y) = \begin{cases} 0 & \text{if } I_1\text{(x,y)=i and } \exists\, (x1, y1) : \text{(x1} \le \text{XB1 or} \\ & \text{x1} \ge \text{XB2 or y1} \le \text{YB1 or y1} \ge \text{YB2) and} \\ & I_1\text{(x1,y1)=i} \\ \text{I(x,y)} & \text{otherwise} \end{cases} \tag{9}$$

2.2. Noisy text region detection and removal

In this stage, we detect noisy text regions of the image $I_c(x,y)$ that resulted from the previous stage. The flowchart of our algorithm is shown in Fig. 5. Before it, we proceed to skew correction based on [8]. Our aim is to calculate the limits, *XT1* and *XT2*, of the text region as shown in Fig. 6. We first proceed to an image smoothing in order to connect all pixels that belong to the same line. Then, we calculate the vertical histogram in order to detect text zones. Finally, we detect noisy text regions with the help of the signal cross-correlation function. The main modules of the proposed technique are described in detail as follows.

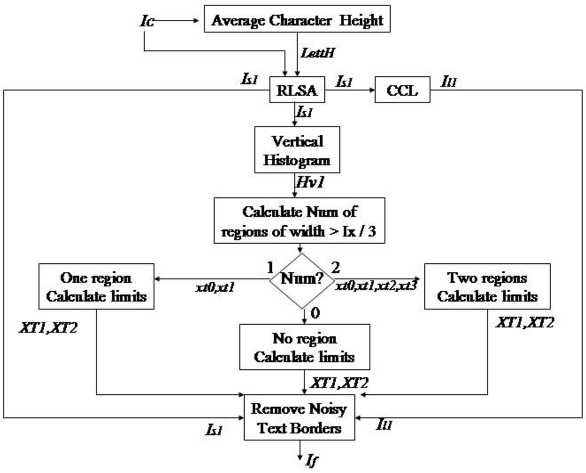

Ic
Average Character Height
LettH
Is1
RLSA
Is1
CCL
Il1
Is1
Vertical Histogram
Hv1
Calculate Num of regions of width > Ix / 3
1
2
Num?
One region Calculate limits
xt0,xt1
xt0,xt1,xt2,xt3
Two regions Calculate limits
0
XT1,XT2
XT1,XT2
No region Calculate limits
XT1,XT2
Remove Noisy Text Borders
Is1
Il1
If

Figure 5. Flowchart for noisy text region detection and removal.

Average Character Height: The average character height (*LettH)* for the document image is calculated based on [9].

RLSA: Horizontal and vertical smoothing with the use of the RLSA [6] by using dynamic parameters which depend on average character height (LettH). Our aim is to connect all pixels that belong to the same line. The horizontal length threshold is experimentally defined as the LettH while the vertical length threshold is experimentally defined as 50% of the LettH. The resulting image is $I_{s1}(x,y)$.

CCL (Connected Component Labeling): Extract the connected components of the image [7]. The image consists of CS connected components C_i and the resulting image is $I_{l1}(x,y)$ as in Eq. (2).

Vertical Histogram: Calculate vertical histogram H_{v1} as follows:

$$H_{v1}(x) = \sum_{y} I_{s1}(x,y) \qquad (10)$$

Calculate Number of regions of width $> I_x^{'}/3$: Check if the number of consecutive *x*, where $H_{v1}(x) > L_4$, is greater than $W = (1/3) * I_x^{'}$ (where $I_x^{'} = XB2 - XB1$) and calculate the number of regions that satisfy this condition. Let suppose that two regions have been found and let xt_0, xt_1 and xt_2, xt_3 denote the regions' limits, as shown in Fig. 7. Similarly, if one region has been found we set xt_0, xt_1 to denote the region's limits.

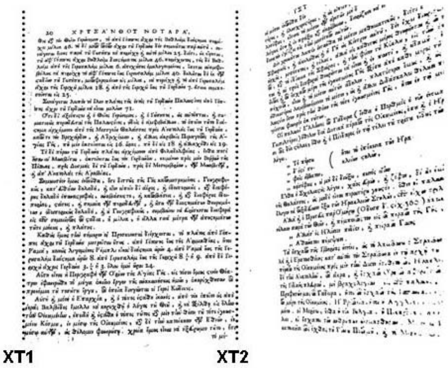


Figure 6. Limits XT1 and XT2 of text region after noisy text region detection.

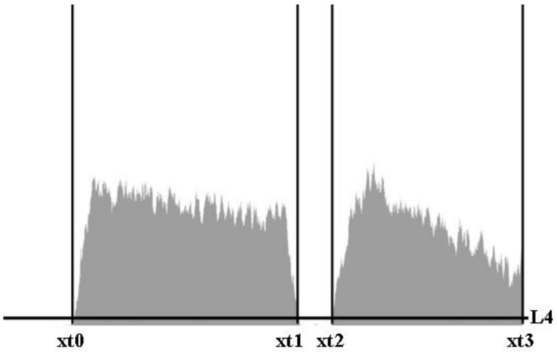


Figure 7. Projections of image in Fig. 5 and text regions detection.

Two regions-Calculate Limits: We examine if one of these regions is a noisy text region. Calculate signal cross-correlation for each region (SC_0, SC_1) [10]. First, we calculate SC_y (Eq. 11) for each line of the region

$$SC(a,y) = 1 - \frac{2}{M}\sum_{k=0}^{M}(I_{s1}(k,y)\ \mathrm{XOR}\ I_{s1}(k,y+a)) \qquad (11)$$

where M is the region's width and *a* is the distance between two lines. Finally, total SC_i of region *i*, is the middle count of all SC_y.

Then, we calculate limits *XT1* and *XT2* as follows:

```
if ($SC_0 < 0.5$ AND $SC_1 < 0.5$) then
  ($XT1 = xt0$ AND $XT2 = xt3$)
else if ($SC_0 < SC_1$) then
    ($XT1 = xt0$ AND $XT2 = xt1$)                (12)
else
    ($XT1 = xt2$ AND $XT2 = xt3$)
```

One region-Calculate Limits: We examine if the noisy text region and the text region are very close to each other without leaving a blank line between them. If the width of region is less than 70% of $I_x^{'}$ we consider that we don't have noisy text region, so $XT1 = xt_0$ *and* $XT2 = xt_1$. Otherwise, we divide it into eight regions and calculate the signal cross-correlation for each region ($SC_1,...,SC_8$) using Eq. 11.

Calculate XT1 and XT2 as follows:

- If $SC_1 < 0.5$ and $SC_8 < 0.5$ we don't have noisy text region, so $XT1 = xt_0$ and $XT2 = xt_1$.
- If $SC_1 > 0.5$ we search for the last consecutive region *i* where $SC_i > 0.5$ and we find an x' where H_{v1} is minimum in this region.
  If $(xt_1 - x') \ge W$ then
    $XT1 = x'$ and $XT2 = xt_1$
  else
    $XT1 = xt_0$ and $XT2 = xt_1$
- If $SC_8 > 0.5$ we search for the last consecutive region *i* where $SC_i > 0.5$ and we find an x' where H_{v1} is minimum in this region.
  If $(x' - xt_0) \ge W$ then
    $XT1 = xt_0$ and $XT2 = x'$
  else
    $XT1 = xt_0$ and $XT2 = xt_1$

No region-Calculate Limits: In this case, the text region consists of two or more columns and we try to locate and separate them from the noise text regions, if these exist. First, we check if the number of consecutive x, where $H_{v1}(x) > L_4$, is greater than $W/4$. If two or less regions are find that satisfy the conditions, we set *XT1=XB1* and *XT2=XB2*. If we find three or more regions that satisfy the conditions we calculate the signal cross-correlation (Eq. 11) for the left and the right region (SC_0, SC_1). Consider that left region's limits are xt_0 and xt_1 and the right region's limits are xt_2 and xt_3. We calculate *XT1* and *XT2* as in Eq. 12.

Remove Noisy Text Region: All black pixels that belong in a connected component C_i which does not include at least one pixel in the limits *XT1*and *XT2* are transformed in white. The final image $I_f(x,y)$ is calculated as follows:

$$I_f(x,y) = \begin{cases} I_c(x,y) \;\; I_{l1}(x,y) = i \text{ AND } \exists\, (x1,y1): \\ \qquad (x1 \ge XT1 \; or \; x1 \le XT2) \text{ AND } I_{l1}(x1,y1) = i \\ 0 \qquad \text{otherwise} \end{cases} \quad (13)$$

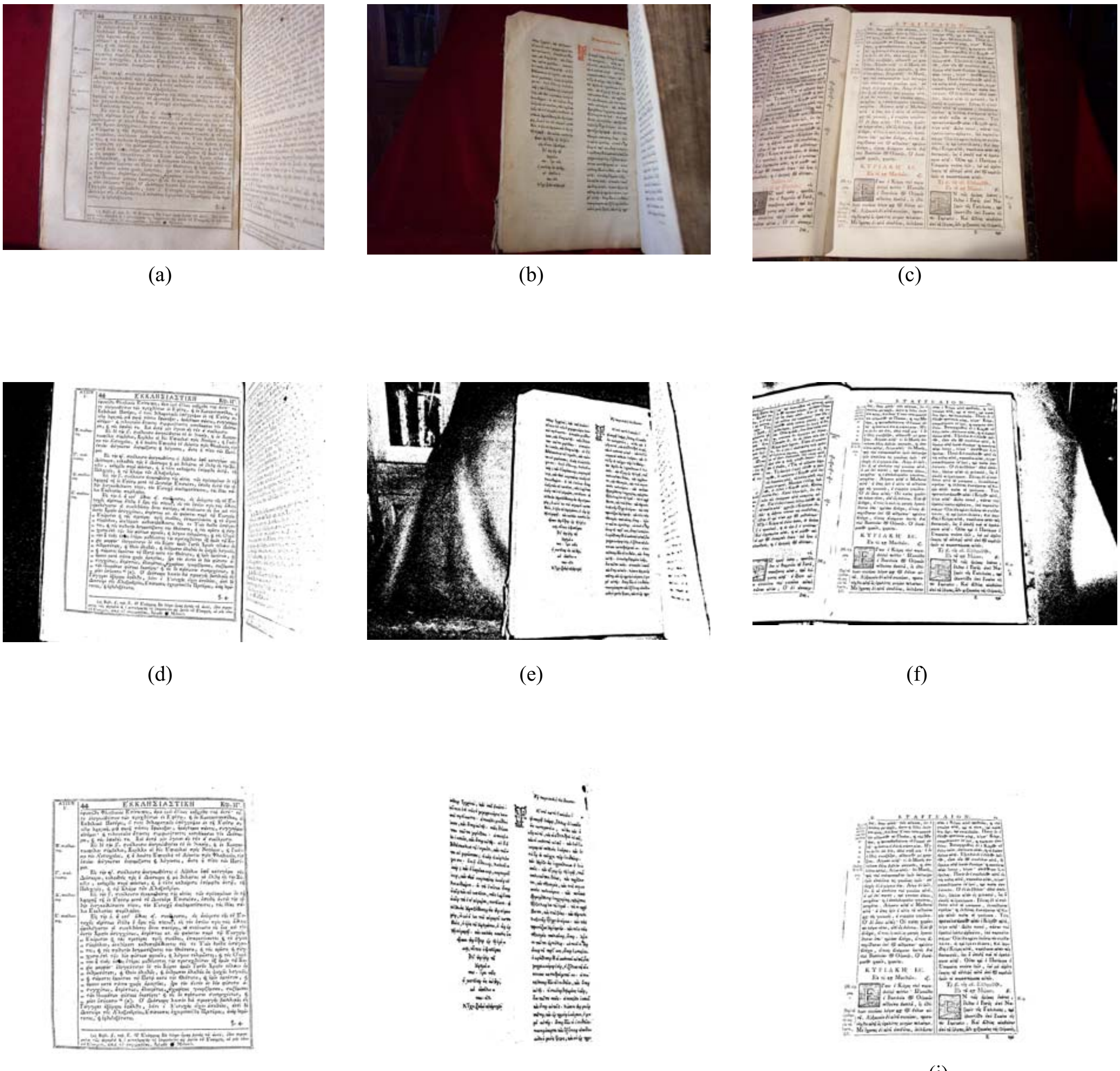
3. Experimental results

To verify the validity of the proposed method, experiments were conducted on several camera document images. We used 1705 document images mainly consisting of historical documents that contain noisy black borders as well as noisy text region appearing from adjacent pages. After visual checking, we found that in 1344 images (78,82% of testing set) the noisy black borders and the noisy text region were correctly removed. Fig. 8 depicts some examples of document images illustrating the page borders detection and removal processes. Difficulties arise in two cases. First, the text region and the noisy text region may be very close to each other without any blank line between them. In this case, a part or even a whole noisy text region may still remain in the resulting image. The second, and perhaps even more difficult case, is when the noisy black border merges with the text region. This may lead to loss of information.

In order to compare our approach with other state-of-the-art approaches, we implemented the methods of Fan et al. [1], Avila and Lins [3] (invading and non-invading algorithms) and used the implementation of the recent algorithm of Avila and Lins [4] found in [11] (see Figs. 9,10). All these methods have been proposed to remove only noisy black borders and not noisy text regions. In the first example we see that only Fan's method and the invading algorithm can effectively remove the noisy black border. Moreover, in the second example, in which noisy black borders are not continuous, none of these methods can effectively remove it.

4. Conclusion

This paper proposes an effective algorithm for the detection and removal of noisy black borders and noisy text regions inserted on document images that are captured by a digital camera. The new algorithm is based on projection profiles combined with a connected component labelling process. Additionally, signal cross-correlation is used in order to verify the detected noisy text areas. Experimentations in several camera document images prove the effectiveness of the proposed methodology. Our future research will focus on the optimization of the proposed method in the cases, when the noisy black border merges with the text region and when the text region and the noisy text region are very close to each other, as described in section 3.

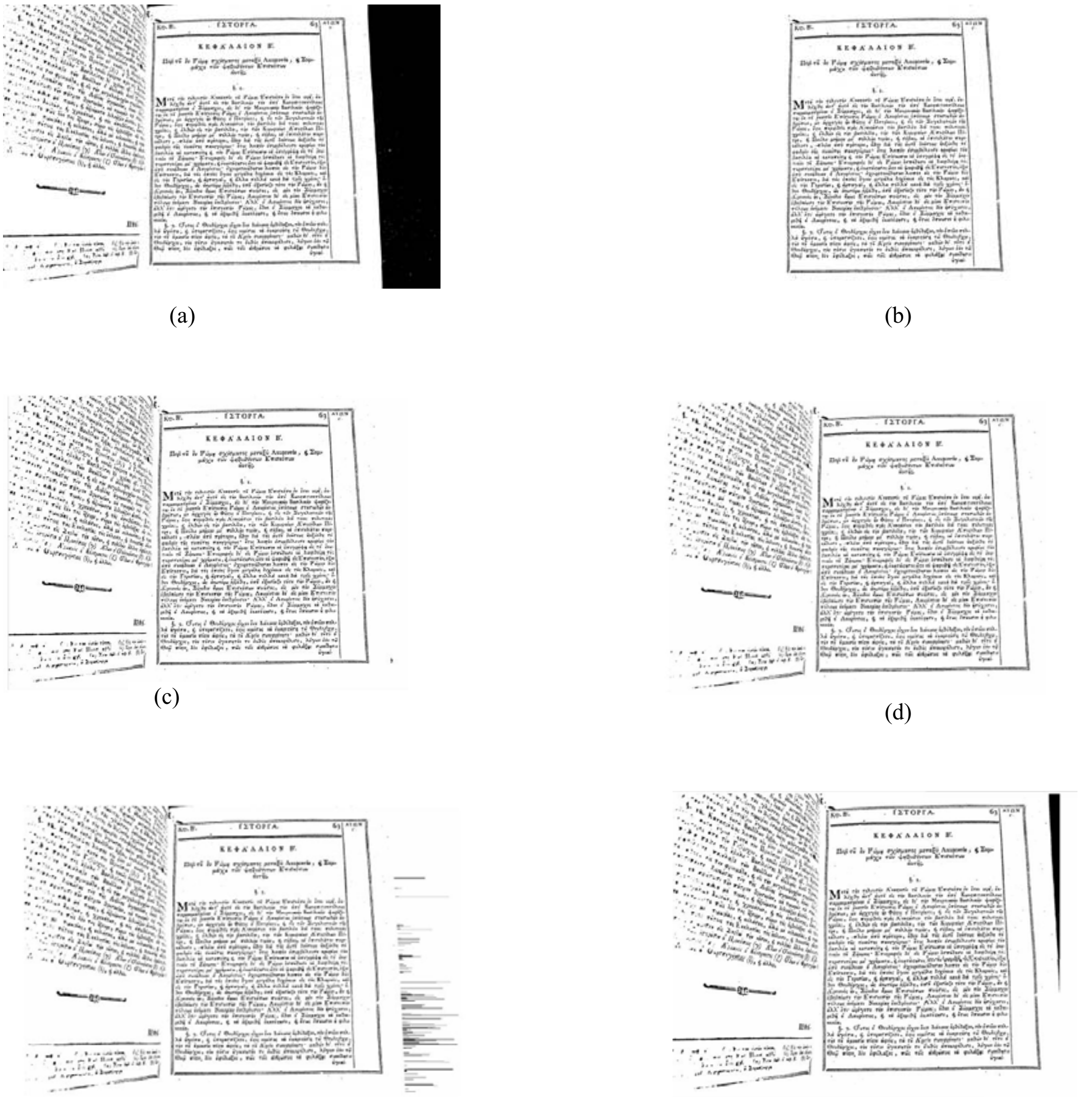


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8. (a)-(b)-(c) Original camera document images, (d)-(e)-(f) binary images, (g)-(h)-(i) results of proposed method.



(a)

(b)

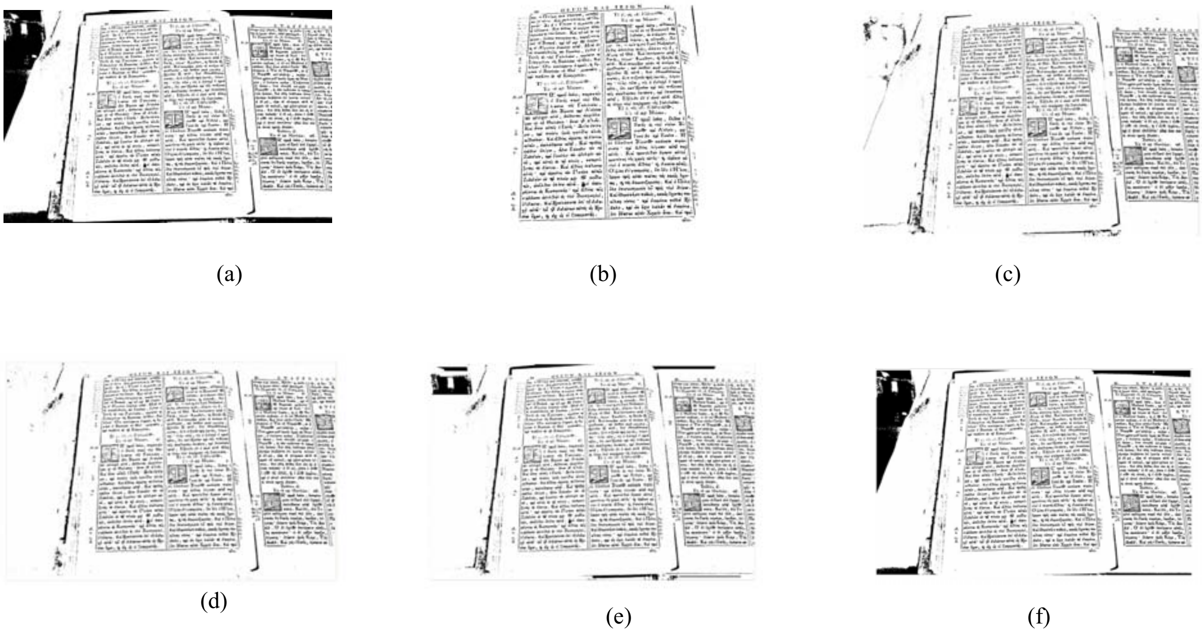
(c)

(d)

(e)

(f)

Figure 9: (a) Original image (b) proposed method (c) method of Fan et al. (d) invading algorithm (e) non-invading algorithm (f) an approach of the algorithm of Avila and Lins.



(a) (b) (c)

(d) (e) (f)

Figure 10: (a) Original image (b) proposed method (c) method of Fan et al. (d) invading algorithm (e) non-invading algorithm (f) an approach of the algorithm of Avila and Lins.

5. References

[1] Kuo-Chin Fan, Yuan-Kai Wang, Tsann-Ran Lay, “Marginal Noise Removal of Document Images”, Pattern Recognition, 35(11), 2002, pp. 2593-2611.

[2] D.X. Le, G.R. Thoma, “Automated Borders Detection and Adaptive Segmentation for Binary Document Images”, International Conference on Pattern Recognition, 1996, p. III: 737-741.

[3] B.T. Avila and R.D. Lins, “A New Algorithm for Removing Noisy Borders from Monochromatic Documents”, Proc. of ACM-SAC’2004, Cyprus, ACM Press, March 2004, pp 1219-1225.

[4] B*.T. Avila, R.D. Lins, “*Efficient Removal of Noisy Borders from Monochromatic Documents”, ICIAR 2004, LNCS 3212, 2004, pp. 249-256.

[5] B. Gatos, I. Pratikakis and S. J. Perantonis, "Adaptive Degraded Document Image Binarization", *Pattern Recognition*, Vol. 39, 2006, pp. 317-327.

[6] Wahl, F.M., Wong, K.Y., and Casey R.G.: “Block Segmentation and Text Extraction in Mixed Text/Image Documents”, Computer Graphics and Image Processing, 20, 1982, pp 375-390.

[7] Fu Chang, Chun-Jen Chen, Chi-Jen Lu, “A linear-time component-labeling algorithm using contour tracing technique”, *Computer Vision and Image Understanding, Vol. 93, No.2,* February 2004, pp. 206-220.

[8] B. Gatos, N. Papamarkos and C. Chamzas, “Skew detection and text line position determination in digitized documents”, *Pattern Recognition*, Vol. 30, No. 9, 1997, pp. 1505-1519.

[9] B. Gatos, T. Konidaris, K. Ntzios, I. Pratikakis and S. J. Perantonis, “A Segmentation-free Approach for Keyword Search in Historical Typewritten Documents”, *ICDAR*, Seoul, Korea, August 2005, pp. 54-58.

[10] Sauvola J., Pietikainen, M.: “Page segmentation and classification using fast feature extraction and connectivity analysis”, ICDAR, 1995, pp. 1127-1131.

[11] Software of Personal PC Helpers (http://www.sharewareconnection.com/bordershelper.htm).

Tableau - Processing Teaching-board Images Acquired with Portable Digital Cameras

Daniel Marques Oliveira and Rafael Dueire Lins
Departamento de Eletrônica e Sistemas – UFPE – Recife – PE – Brazil
danielmarquesoliveira@gmail.com, rdl@ufpe.br

Abstract

Portable digital cameras are of widespread use today due to good image quality, low cost and portability. Teaching-boards are the most universal classroom equipment throughout the world. This paper presents a software environment for processing images from teaching-boards acquired using portable digital cameras and cell-phones.

Keywords: *Digital cameras, image processing, portable cameras, teaching boards.*

1. Introduction

Portable digital cameras were developed for taking amateur "family photos"; the recent price-performance improvement, low weight, portability, low cost, small dimensions, etc. widened enormously the number of users of digital cameras giving birth to several new applications. One of them, completely unforeseen is using portable digital cameras for digitalizing images from teaching-boards. Teaching boards are present in every classroom throughout the world with small variations: modern ones are white and written with color felt tip markers; some others are black or green and written with white chalk sticks. Students take notes of what teachers write on the board for later revision. Today, some students start to take photos of classroom boards for later reference.

This paper describes a software environment to process images of teaching boards acquired using portable digital cameras operated either by students or teachers. This simple tool provides a natural way to generate digital content for courses, respecting particular aspects of the group such as syllabus, class learning speed, teacher experience, regional content, local culture, etc.

The system consists of three parts. The first is database formation. As soon as the images are transferred from the camera to the PC information is collected to generate a simple database that will organize the images for later content formation. Information such as teacher name, course name, discipline, subject, class number, group number, etc. are requested. The second module is for image processing. This module will improve the image acquired in a number of ways involving background removal, image segmentation, skew correction, image enhancement, etc. The third part of the processing environment deals with outputting the content. Three different ways are under development: printed handouts, webpage generation and slide production. Each of these media receives the information of the processed image part of the environment and makes it suitable to its best use. This paper focuses on the image processing parts of the environment.

2. Image Acquisition

Image acquisition is performed by taking a photograph of the teaching board at a suitable distance, before cleaning up the information. Whenever a photo is taken, special care is needed to keep the readability of the text in the inbuilt camera LCD display. The image processing part takes the images acquired by a portable digital camera and processes them in a number of ways. Very often the photograph goes beyond the board size and incorporates parts of the wall that served as mechanical support for taking the photo of the document. Boards often have frames either made of wood or metal. The second problem is due to the skew often found in the image in relation to the photograph axes, as cameras have no fixed mechanical support very often there is some degree of inclination in the document image. The third problem is non-frontal perspective, due to the same reasons that give rise to skew. A fourth problem is caused by the distortion of the lens of the camera. This means that the perspective distortion is not a straight line but a convex line (in digital camera photos) or concave line (in cell phone photos), depending on the quality of the lens and the relative position of the camera and the document. The fifth difficulty in processing board images acquired with portable cameras is due to non-uniform illumination. White boards have a polished surface to avoid the marker ink to be absorbed by the board surface. This yields non uniform photo illumination as one often finds high intensity bright areas that correspond to reflections of room lighting. Figure 1 presents an example of a white board photographed with a low-cost mobile phone Nokia 6020, where one may observe the four problems aforementioned: extra borders, image skew, non-frontal perspective distortion, and lens distortion. Besides those problems one may add: uneven multiple source illumination and non-delimited areas. One must remark that all pictures taken for this study and presented herein were from real classes. In the case of the board presented in Figure 1, there is a written area in the top-leftmost area that belonged to a "previous" board area. The lecturer did not respect the board junction that imposes a natural area delimiter. The pictures were taken after lectures without previous information to the lecturer. If informed beforehand, lecturers ought to respect area separation to make easier board segmentation. What is most surprising is that despite the low resolution of the camera of the cell-phone and the non-ideal environment, the image obtained provides readable information.

The board image presented in Figure 2 also exhibits the four problems already described. The photo was taken without natural daylight interference and strobe flash (the HP iPaq has no internal strobe flash). Room illumination was from tube fluorescent lamps. One may notice that the lecturer respected the board junction as a content delimiter.

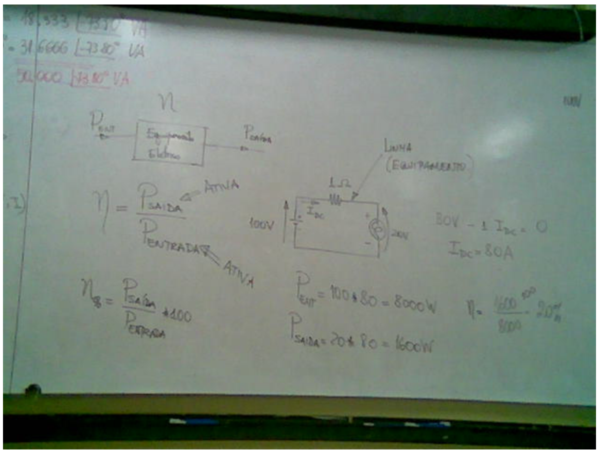


Figure 1. Part of a white-board acquired with internal camera of the cell-phone Nokia 6020, no strobe-flash used, 640x480 pixels, image size 21KB under Jpeg compression, board height 115cm, illumination: natural daylight (indirect) and ceiling tube fluorescent lamps.

An Olympus portable digital camera was used to acquire the board image presented in Figure 3. Two aspects are new in this image. The first is the presence of some blurred areas due to imperfect board cleaning or aging of surface of the board. The second is that the lecturer used vertical lines to split the content of his presentation on different board segments.

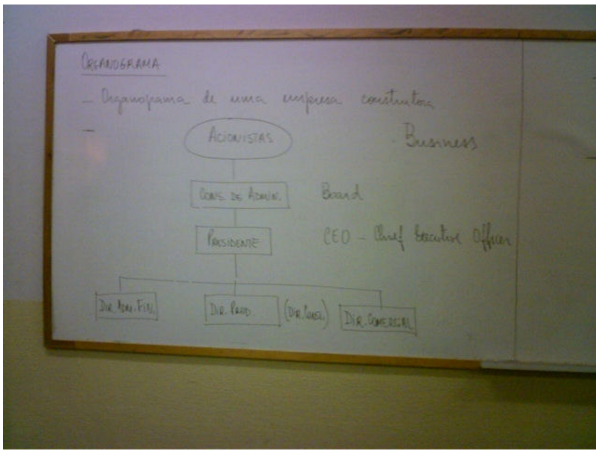


Figure 2. Part of a white-board acquired with internal 1.2 Mpixel camera of a HP iPaq rx3700, no strobe-flash used, image size 131KB under Jpeg compression, board height 115cm, illumination: natural ceiling tube fluorescent lamps.

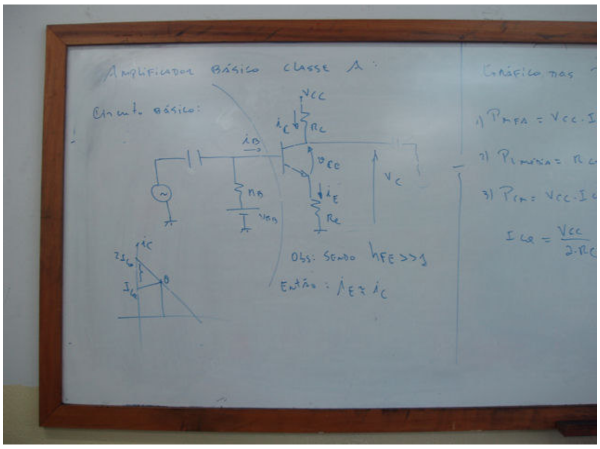


Figure 3. Part of a white-board acquired with portable digital camera Olympus C60, 6.0 Mpixels, no strobe-flash used, image size 1.20MB under Jpeg compression, board height 115cm, illumination: natural ceiling tube fluorescent lamps.

3. Boundary detection

The first step to process teaching board information is to find the limits of the image. As already mentioned, boards often have frames as a decorative "finishing" or mechanical support as may be observed in figures 2 and 3, but that is not always found. Figure 1 is an example of the latter case. Besides that, in real classrooms a board is several meters wide. Thus, the content of a board often claims for several photos to be covered. Figures 2 and 3 exemplify a left section of a board while Figure 1 presents a central slice of a board. One should observe that the non-framed edges bring a higher-complexity for boundary detection, thus for image segmentation.

Boundary detection is central for all other steps described herein because it isolates the area of interest from its surroundings. Besides that, the detection of the boundaries will allow one be able to correct perspective and skew in the image. Unfortunately, boundary detection has shown itself a much harder task than one may first assume. The algorithm presented in reference [8] [9] used to remove the mechanical background from document images acquired with portable digital cameras is unsuitable for board images, despite the high degree of similarity in the two problems addressed. A new algorithm is presented herein, performing the following steps:

3.1 Segmentation

1. Split the input image (ORI_IMG) into 4 regions as presented in Figure 7
2. Create a new binary image (DIF_IMG) of equal dimensions;
3. H_DIST and V_DIST are defined as functions of the image resolution. They correspond to the rounding-off of the integer part of 0.91% of the width and height in pixels of the original image, respectively. For instance, in the case of a 640x480 pixel image, H_DIST=6 and V_DIST=4.
4. DIF_IMG(x,y) is white if one of the following condition holds, it is black otherwise:

- The difference between each component of ORI_IMG(X,Y) and ORI_IMG(X±H_DIST,Y±V_DIST) is less than 10
- The componentwise gap of ORI_IMG(X,Y) and ORI_IMG(X±2.H_DIST,Y±2.V_DIST) is < 10.
- The pixel differences are local operations, which minimize non-uniform illumination. A difference between non-board areas and board areas is more likely to turn black in DIF_IMG, than if all pixels compared belong to the board.

Two pixels differences are needed to minimize the "double contour" around board writings. The first contour is marked as black when ORI_IMG(X,Y) is located on the teacher writing and the inner pixel is a board background. The second contour is the other way round, thus DIF_IMG is wrongly marked as black. Such behavior may be seen in Figure 5 for the letter "A" obtained from Figure 3. The behavior when considered the two differences is shown in Figure 6.

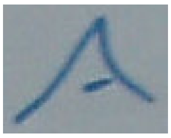


Figure 4. Letter "A" extracted from Figure 3

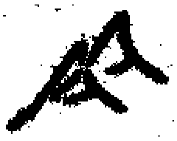


Figure 5. Segmentation considering just one difference

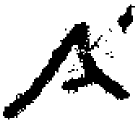


Figure 6. Segmentation considering the two differences

The sign is such as always to subtract the outermost value from the innermost one, according to the matching position from the region on Table 1.

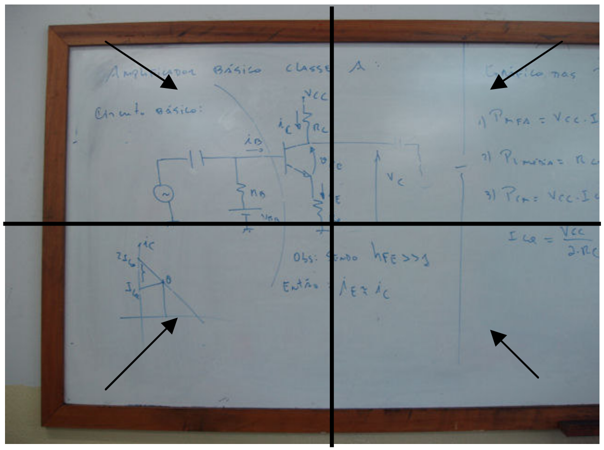


Figure 7. Board split into four regions. Arrows show the direction of illumination compensation.

The result of Step 1 applied to the image presented in Figure 7 is shown in Figure 8.

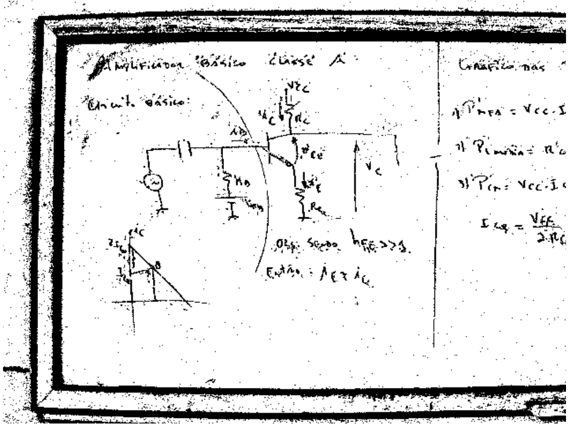


Figure 8. Figure 3 board image after Step 1.

Table 1. H_DIST and V_DIST offset calculation.

X+H_DIST, Y+V_DIST	X-H_DIST, Y+V_DIST
X+H_DIST, Y-V_DIST	X-H_DIST, Y-V_DIST

3.2 Finding Control Points

Points that possibly belong to the board boundaries are called control points. This step will try to spot them by analyzing the binary image.

For each direction, N equally spaced axes are defined to scan the binary image looking for control points from the centers towards the borders. Each of those vertical axes is swept with a 7x1 pixel mask. Similarly, each horizontal axis is scanned with a 1x7 pixel mask. If four or more pixels under the mask are black, go to 2. For the collected images N was set up to 9.

1. Calculate colors ratio in the rectangle around current point (X, Y). The bottom border of the rectangle has the upper left corner located on (A,B) and the lower right corner on (C,D), where:

$$A = X - \left[\frac{IMG\_WIDTH}{N} \times \frac{1}{2}\right]$$

$$B = Y - IMG\_HEIGHT \times INTERNAL\_CHECK$$

$$C = X + \left[\frac{IMG\_WIDTH}{N} \times \frac{1}{2}\right]$$

$$D = Y + IMG\_HEIGHT \times EXTERNAL\_CHECK$$

   * Where INTERNAL_CHECK=0.165% and EXTERNAL_CHECK=0.91%

2. If within the rectangle (A,B,C,D) there is more than 65% of black pixels, then (X,Y) is marked as a control point candidate for the bottom border. Otherwise, the algorithm moves outwards looking for another candidate.

For all other directions the algorithm works similarly to the step explained above. An example of control points found for the board image in Figure 8 is presented in Figure 9.

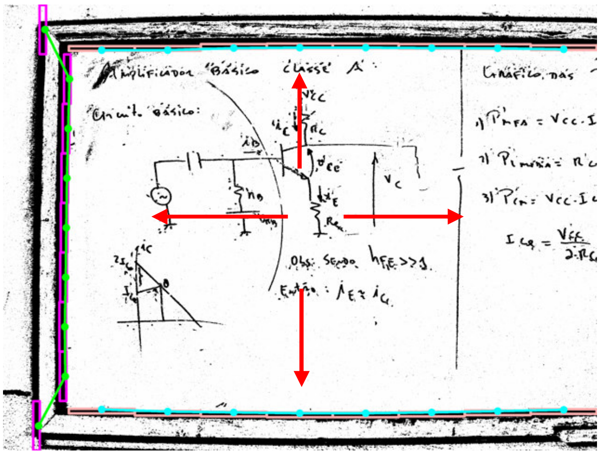


Figure 9. Board image from Figure 8 scanned in red from the center towards the outwards exhibiting the control points on top-bottom-left positions.

3.3 Control point elimination

Control points can be wrongly found, due to:

- The ORI_IMG(X,Y) it is out of the boundary of the board, as shown in Figure 9;
- When the board has added "noise" such as a sign or advertisement;
- The board picture was taken with strobe flash, etc.

To eliminate such points the following procedures are executed:

1. For every control point candidate, tangents of the angles (Θn) with the border axis formed by the candidate and its 2-neighborhood in both sides are calculated. A candidate is selected if the absolute value of at least 2 tangents is lower or equal to 0.07. One may observe that this calculation is not relative to the number of the neighborhood, so if the candidate is the outermost point all tangents should be lower or equal to 0.07. An example of a horizontal border neighborhood is shown in Figure 10, where the candidate is in dark grey.

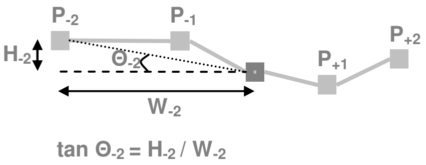


Figure 10. Control point selection

1. After executing step 1 above in all directions, the outermost points will define a line segment as depicted in Figure 11. Any candidate to a control point outside the orthogonal segment defined is excluded.

Figure 12 shows control point candidates. Figure 13 shows the image after the deletion of the wrong ones.

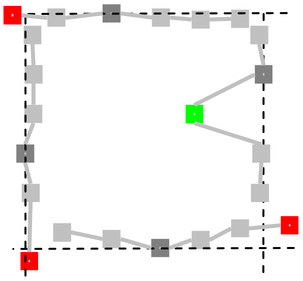


Figure 11. Green CP is eliminated in step one, while red ones are eliminated in step two.

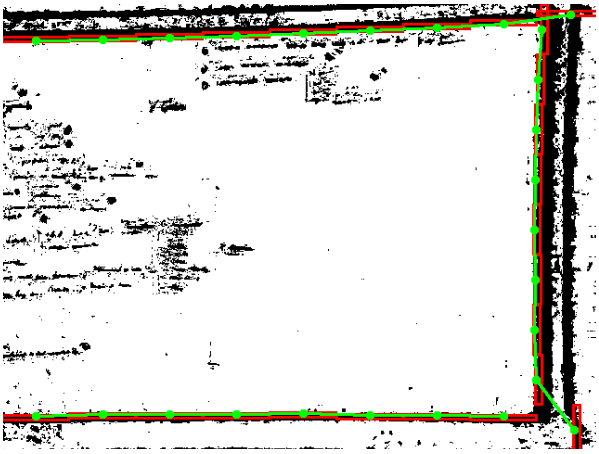


Figure 12. Board image showing two wrong control points that are deleted.

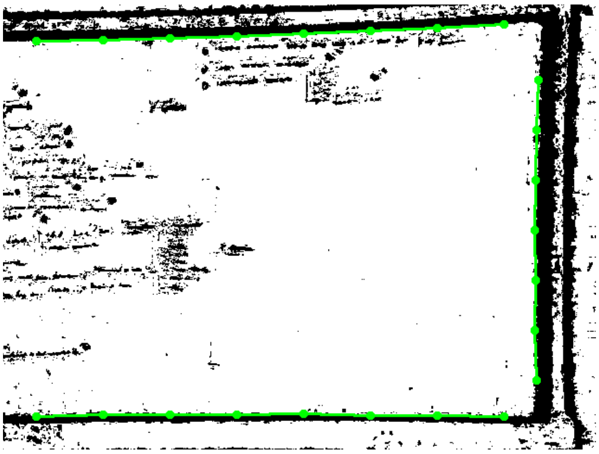


Figure 13. Board image after control point elimination.

4. Perspective Correction and Cropping

The freedom allowed in acquiring board images with portable digital cameras without mechanical support invariably leads to perspective distortion. Perspective transformation must be done in order to make the image appear as a frontal vision. This process is called rectification [1] [3] [4] [5] [6] [10]. Four edges that delimit the original board image are needed. There are four kinds of board images:

1. The image presents no lateral borders as presented in Figure 1.
2. The image presents a left border on the image, as shown in Figures 02 and 03.
3. The image presents a right border on the image.
4. The whole board image fits the photograph.

In any of the cases above four points were taken as reference for perspective correction. Those points were chosen by drawing a line passing through the two outermost points in each direction and finding their intersections, which are named the *reference points*. If no control point is found in any direction the intersection of the lines drawn with the end of the image is taken as a reference point. This often happens in the three first cases above.

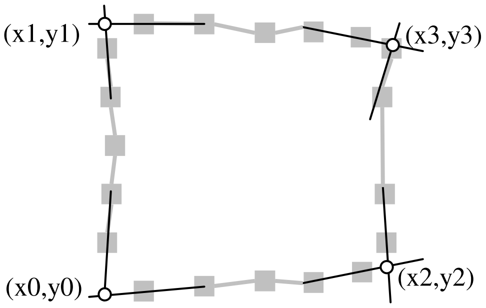


Figure 14. Reference points for perspective correction

One must remark that the technical literature registers other perspective correction techniques in the absence of reference points [4]. The adoption of the choice of reference points as above was done for a matter of uniformity and simplicity, and provided good results as is discussed later on.

Once the four reference points are chosen their equivalent after perspective correction are calculated as:

```
d1=|x0-x2|+|y0-y2|; d2=|x1-x3|+|y1-y3|;
d3=|x0-x1|+|y0-y1|; d4=|x2-x3|+|y2-y3|;
aspect=(d1+d2)/(d3+d4); x'0=x'1=x0;
x'2=x'3=xd0+d1; y'0=y'2=y0;
y'1=y'3=y'0+(d1/aspect);
```

5. Image Enhancement

There are several algorithms in the literature for enhancing images. Image board enhancement has to increase the contrast between the teacher writings and the board background, increasing information readability. Finding a way to cluster similar information to widen the gap between the different clusters is the key to image enhancement in the case of teaching-board images.

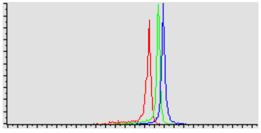


Figure 15. Background histogram of Figure 04

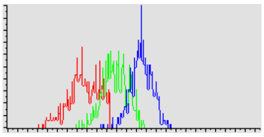


Figure 16. Histogram of Letter of Figure 04

Figure 15 and Figure 16 present the histograms of different selections of Figure 04. The letter “A” has a more representative contribution of the blue component, and the color-histogram is widespread. The background histogram is narrower and more uniform. An efficient way to increase the contrast, without affecting the feature of an image, is provided by Rayleigh filter [7].

The images were tested against the following classes of algorithms: global and local histogram equalizations sharpen, and mathematical morphology techniques [7]. The Rayleigh filter with parameter λ= 0.5 consistently provided the best results. Figures 17 to 19 show images after Tableau processing.

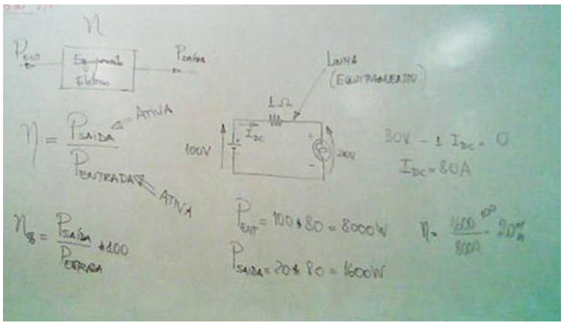


Figure 17. Tableau applied to Figure 1

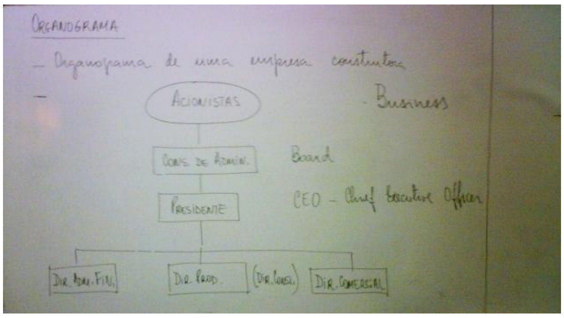


Figure 18. Tableau applied to Figure 2

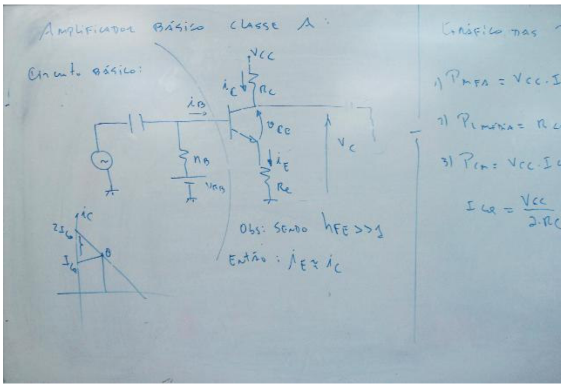


Figure 19. Tableau applied to Figure 3

6. Tableau in ImageJ

ImageJ [11] is an open source image processing environment in Java developed by Wayne Rasband, is at the Research Services Branch, National Institute of Mental Health, Bethesda, Maryland, USA.It allows the insertion of plug-ins for special purposes. The algorithms presented herein for processing teaching board images was developed as an ImageJ plugin. Figure 20 presents a screen shot of the Tableau interface menu.

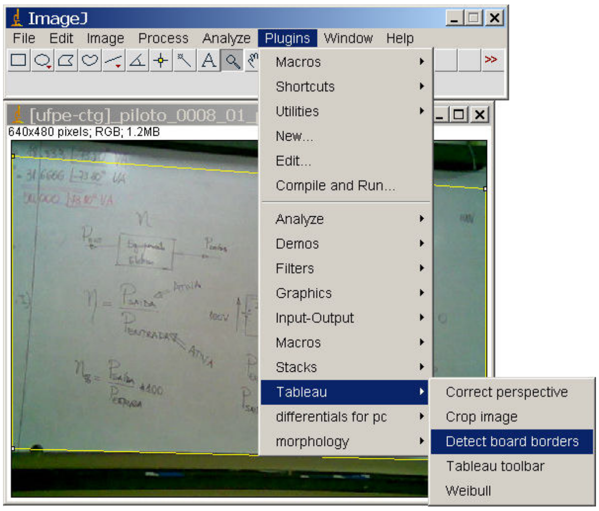


Figure 20. Tableau Plug-in interface in ImageJ

The algorithm for border detection presented above sometimes does not yield the best choice. The Tableau plug-in in ImageJ allows the user to adjust the contour detection for better processing.

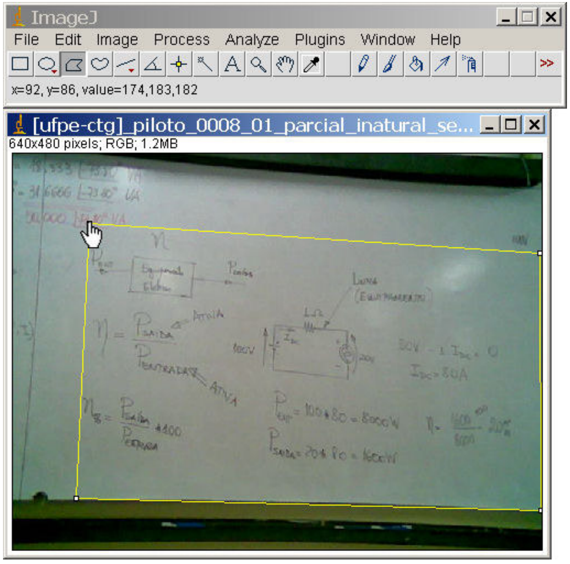
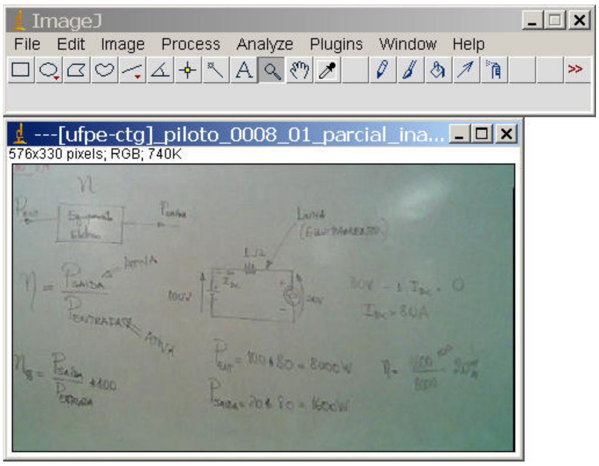

ImageJ
File
Edit
Image
Process
Analyze
Plugins
Window
Help
x=92, y=86, value=174,183,182
[ufpe-ctg]_piloto_0008_01_parcial_inatural_se...
640x480 pixels; RGB; 1.2MB

Figure 21. Boundary corrected for Figure 1.



---[ufpe-ctg]_piloto_0008_01_parcial_ina...

576x330 pixels; RGB; 740K

Figure 22. Perspective corrected and cropped image from Figure 21.

Figure 20 presents the automatic boundary selection performed by the algorithm in the background, while Figure 21 shows the operator driven selection for the same image. The perspective corrected/cropped image is in Figure 22. Image in Figure 1 has size of 27.5 Kbytes while the cropped image claims only 15Kbytes of storage space, both compressed in JPEG. Average times for each of the board processing phases are showed on Table 2. They were measured considering the end-user perspective, thus screen refresh and progress bar updates are included in processing times.

Table 2. Average processing times in ImageJ

Tableau	6.0 Mpixel	1.2 Mpixel	300 Kpixel
Border detection	361ms	104ms	22ms
Persp. correction	42857ms	9692ms	2187ms
Image crop	589ms	224ms	79ms
Rayleigh ($\alpha = 0.5\sqrt{2}$)	515ms	198ms	60ms
Total time	44322ms	10218ms	2348ms

Times were measured on processor AMD Sempron 2600+ 1.83 GHz with 512Mb RAM running on Windows XP SP2. One may see that perspective correction in Tableau is very time consuming. This is due to the use of the JAI (Java Advanced Imaging) version 1.1.3 with machine native support [12] which is not incompatible with ImageJ, demanding to-and-from conversion of representations of the two libraries. The development of an ImageJ plug-in for such purpose will certainly yield more efficient code. Bicubic2 interpolation with subsampled bits equal to two was used [1]. One may see that the Rayleigh algorithm takes longer to process than the Border detection algorithm. This is due to the image refreshing window needed by Rayleigh while detection only shows the selection of board corners to the user.

Tableau was tested on 81 images from Olympus C60, 53 from Nokia 6020 and 3 from HP iPaq rx3700.

7. Comparisons with other approaches

Tableau was compared with two document processing web environments: Qipit®[13] and ScanR®[14]. The former was tested with 16 Olympus images, 3 from HP iPaq and 6 from Nokia 6020. The latter was tested on a subset of those images, as ScanR does not handle low-resolution images such as the ones taken with the Nokia 6020. Typical comparative results for the board of Figure 23 may be seen in Figures 24 to 26, working in similar circumstances.

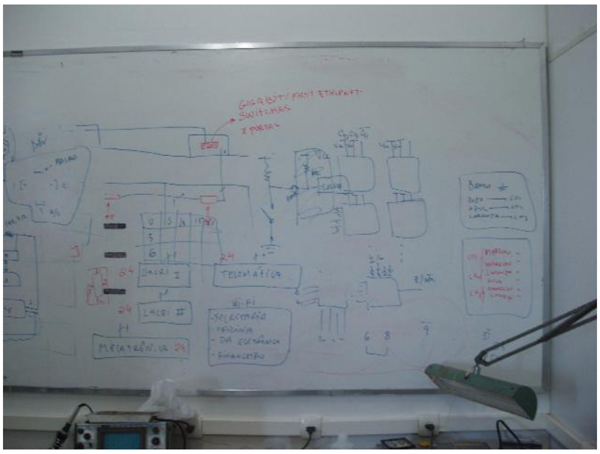


Figure 23. Original image

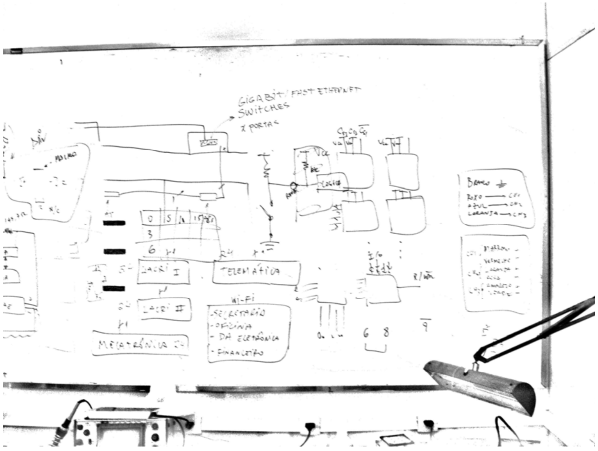


Figure 24. Qipit® processing

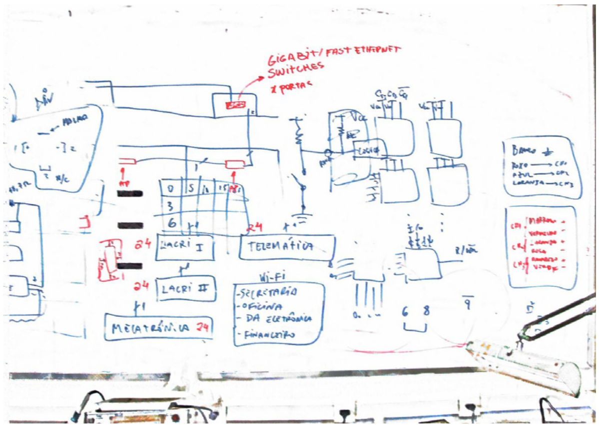


Figure 25. ScanR® approach

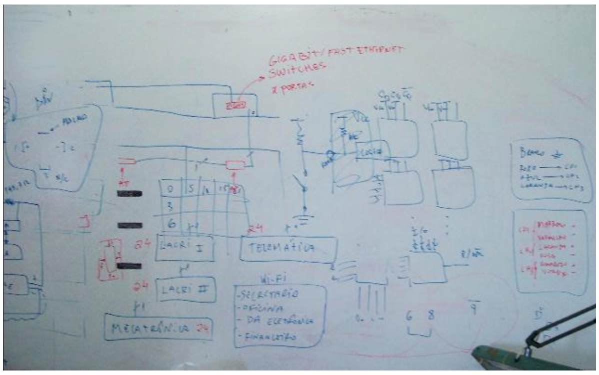


Figure 26. Tableau approach

As Figure 24 shows, Qipit neither detected the board border nor corrected the perspective, but binarized the image. ScanR, as Figure 25 shows, performs a poor border detection and crop, but a good image enhancement. Tableau, as shown in Figure 26, presents the best border detection and crop of the tools studied, but image enhancement falls behind ScanR. Performance comparison could not be done because both Oipit and ScanR systems are on-line Internet services.

8. Conclusions and Further Works

Portable digital cameras are a technological reality today that opens a wide number of challenges in image processing, including board image processing. This paper presented the image processing part of Tableau, an environment for assisting the layman to easily generate digital contents from teaching-board images. The image processing part of Tableau was implemented in ImageJ and finds the edges of board images, allowing perspective correction and image cropping, thus eliminating wall background and board frames. All images analyzed here were obtained from real classrooms without any special care either from lecturers or from the environment (illumination, special framing or marking, etc.). Different cameras were tested (different manufacturers, resolution, quality, etc.). Whenever compared with Qipit® and ScanR®, Tableau provided the best image segmentation and cropping.

Better board image enhancement and image binarization are currently under work in Tableau.

The Tableau code is freely available at: http://www.telematica.ee.ufpe.br/sources/Tableau.

9. Acknowledgements

The work reported herein was partly sponsored by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazilian Government.

10. References

[1] D. Liebowitz and A. Zisserman. "Metric rectification for perspective images of planes". In Proc. of the Conference on C. Vision and Pattern Recognition, pp 482–488, 1998.

[2] George Wolberg, Digital Image Warping, 1990, pp 129-131, IEEE, ISBN 0-8186-8944-7.

[3] G. K. Myers, R. C. Bolles, Q.-T. Luong, J. A. Herson, and H. B. "Rectification and recognition of text in 3-D scenes". International Document Analysis and Recognition, 7(2+3) pp 147–158, July 2005.

[4] L. Jagannathan and C. V. Jawahar, "Perspective correction methods for camera based document analysis", pp. 148–154, CBDAR 2005/ICDAR 2005, Korea. 2005.

[5] P. Clark, M. Mirmehdi, "Recognizing Text in Real Scenes", IJDAR, Vol. 4, No. 4, pp. 243-257, 2002.

[6] P. Clark, M. Mirmehdi, "Location and Recovery of Text on Oriented Surf", SPIE CDRR VII, pp.267-277, 2000.

[7] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd ed. Prentice Hall, 2001.

[8] R. D. Lins, A. R. Gomes e Silva and G. P. da Silva. Enhancing Document Images Acquired Using Portable Digital Cameras. ICIAR 2007, LNCS, Springer Verlag, 2007.

[9] R. Gomes e Silva and R. D.Lins. Background Removal of Document Images Acquired Using Portable Digital Cameras. LNCS 3656, p.278-285, 2005.

[10] S. J. Lu, *et al*, "Perspective rectification of document etc.," Image and Vision Computing, V(23):541-553, 2005.

[11] ImageJ http://rsb.info.nih.gov/ij/

[12] JAI (Java Advanced Imaging). https://jai.dev.java.net.

[13] Qipit®. http://www.qipit.com

[14] Scanr®. http://www.scanr.com

Enhancement of Camera-captured Document Images with Watershed Segmentation

Jian Fan
Hewlett-Packard Laboratories
jian.fan@hp.com

Abstract

Document images acquired with a digital camera often exhibit various forms of degradation, with some commonly encountered forms being uneven illumination, color shift, and blurry text. To complicate matters, the content of modern documents has become increasingly complex. The potential combination of poor image quality and complex content is very challenging for an image processing algorithm to cope with. Towards this end, we present an image enhancement algorithm based on watershed segmentation. The problem of over-segmentation is alleviated by noise thresholding of gradient magnitude. The segmentation is then used for illumination and color correction, as well as for direct text sharpening. Our results show that the watershed segmentation is more robust to noise and accurate in object boundaries than a direct region growing. We also show that the proposed method works well with both text-only documents and with mixed text/graphical documents.

1. Introduction

Digital cameras possess several unique advantages for document capture. Compact digital cameras, especially camera phones, are convenient to carry around. Professional digital cameras, with resolutions now commonly exceeding ten million pixels, have been widely used for various large-scale book digitization projects, showcasing the non-destructive nature of digital camera capture. However, document capture with digital cameras has many inherent limitations [1]. It is very difficult to project uniform lighting onto a document surface, and this often results in uneven illumination and color shift in the acquired images. For documents captured with handheld compact cameras, text blur is also commonplace. These degradations are of interest to this paper.

For the purpose of correcting non-uniform illumination, an illumination-reflectance model $f(x,y) = I(x,y)R(x,y)$, where $f(x,y)$ is the observed image, $I(x,y)$ is the illuminant and $R(x,y)$ is the reflectance, is commonly used [2,3]. In practice it is further assumed that the scale of the illumination's fluctuation is far larger than the affected objects and that the original document contains flat and white background areas that cover a significant portion of the total document area. A classic method for removing the illuminant is homomorphic filtering [2]. In [3], Pilu and Pollard presented a method with illuminant estimation by block averaging followed by a direct correction of $R(x,y) = f(x,y)/I(x,y)$. In an effort to reduce the possible influence exerted by the dark text pixels, Shih-Chang Hsia et al proposed a more sophisticated method in which averaging is done only on a number of maximum values within a line section [4]. However, the performance of these methods may be directly affected by the parameter of block size. In general, the block size should be larger than all text strokes such that no block will fall completely within a stroke of text. If this were to occur, the estimated illuminant surface may dip in this area, adversely affecting the quality of the text or object. On the other hand, increasing the block size generally reduces the scale of light fluctuation that the algorithm can cope with.

An alternative to fixed blocks is image segmentation. One such method, proposed by Yanowitz and Bruckstein, relies on edge detection [5]. However, edge detection has two major drawbacks: 1) it is sensitive to noise and threshold selection, and 2) it does not guarantee closed boundaries on all perceived regions. These drawbacks may significantly impair the robustness of an application's performance. Another commonly used tool is watershed transform [6, 7]. Watershed transform is a parameter-free segmentation method based on mathematical morphology. Instead of applying watershed transform directly to grayscale images, S. Beucher applied the watershed transform to image gradient [8]. This framework was adopted by Wang et al for scene text extraction [9]. The main drawback of the watershed-based segmentation method is over-segmentation. Fundamentally, the problem is largely due to noise. It may be alleviated by using various noise filtering techniques and appropriate application-specific heuristics. For scene text extraction, Wang et al used a weighted median filter based anisotropic diffusion (WMFAD) for noise filtering and heuristics of background regions. Their two-step region-clustering process utilizes heuristics of the size, location, border length, and mean color and intensity of the regions. In their text extraction step, they further incorporated heuristics of the height-to-width ratios of connected components. The authors caution that their method "is only suitable for square character extraction" [9].

Document image enhancement differs from text extraction and binarization in two major aspects. First, the output of document image enhancement should closely resemble the original hardcopy. In other words, the integrity and appearance of pictorial regions within a document should be preserved. Secondly, document image enhancement should be effective for documents of wide-ranging text alphabets, fonts, and orientations. These two requirements rule out the use of many heuristics derived from certain text properties for the application.

In this paper, we apply the watershed-based segmentation framework to the enhancement of camera-captured document images. The block diagram of the complete algorithm is shown in Figure 1. First, a linear Gaussian filtering is performed. Second, a color gradient magnitude is computed. This is followed by a hard thresholding, which has proved very effective in eliminating over-segmentation in the background regions. In the fourth step, the watershed transform is applied to the gradient magnitude and the background regions are determined. Finally, segmentation is used to estimate the illuminant surface $I(x, y)$ and color correction multipliers, and to guide a spatial selective sharpening. We shall point out that for the enhancement application it is not necessary to have a complete segmentation such as that for text extraction and binarization. In particular, we do not need to identify small background regions isolated within text characters. Under the smoothness assumption of the illuminant surface, a complete surface $I(x, y)$ may be reconstructed with linear interpolation from incomplete data points. The tolerance of incomplete segmentation simplifies the algorithm and improves its robustness.

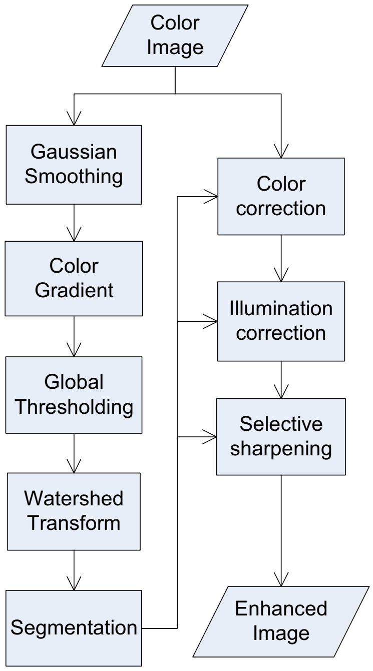


Figure 1. The block diagram of the proposed enhancement algorithm.

The remainder of this paper is organized as follows. Section 2 details the scheme for background segmentation. Section 3 describes image enhancements. Experimental results and comparisons are shown in Section 4. Section 5 summarizes and concludes the paper.

2. Background segmentation via watershed transform

We use the watershed-based framework for the background segmentation. Two major issues are over-segmentation and the identification of the background regions.

2.1. Noise filtering

Since noise is the main source of over-segmentation, noise filtering is critical to the performance of the overall algorithm. Most noise-filtering techniques can be classified as either linear or non-linear. Gaussian filters are commonly-used linear filters. Most edge-preserving and rank filters are non-linear. It has been shown that Gaussian bilateral filtering achieves very similar results as anisotropic diffusion without the complexity of multiple iterations [10, 11]. These filters operate in the image domain. There are others that operate in transform domains. Wavelet shrinkage is a simple and efficient denoising technique in the wavelet domain [11]. For our application, we applied a hard thresholding to gradient magnitude by setting gradient magnitude values below a threshold to zero. The threshold value is a critical parameter. Ideally, it should be determined by background noise level. In practice, the threshold may be set proportional to the standard deviation of the gradient magnitudes:

$$th_g = \begin{cases} k \cdot \sigma_g, & \text{if } (k \cdot \sigma_g) > th_{\min} \\ th_{\min}, & \textit{otherwise} \end{cases}$$

where σ_g is the standard deviation of the gradient magnitudes, k is a real number and th_{min} is a pre-determined minimum threshold value.

Our experiments showed that thresholding of gradient magnitude is significantly more effective than bilateral filtering in reducing over-segmentation of background regions.

2.2. Image gradient

Experiments have shown that better and more complete region boundaries may be detected from color gradient than from grayscale gradient. There have been many color gradient operators proposed in the literature [12, 13]. For this application, we achieved very similar results with either Di Zenzo's color gradient or the simpler "max component gradient." The thresholding operation as described in the last section is then applied to the gradient magnitude to remove the noise component. The denoised gradient magnitudes are then linearly mapped onto an 8-bit grayscale image.

2.3. Background segmentation

After applying Vincent and Soille's fast watershed transform to the 8-bit grayscale gradient image, every pixel is labeled either as a (catchment basin) region or as a watershed. For our application, the watershed pixels are simply merged with the neighboring region possessing the largest label number such that every pixel belongs to a region. The various steps and the effect of gradient thresholding are illustrated in Figure 2. It can be seen that both bilateral filtering (of size 11×11, $\sigma_d = 1.3$ and $\sigma_r = 35$ [10]) and Gaussian filtering (of size 9×9 and $\sigma = 1.3$) fail to completely remove background noise, resulting in severe over-segmentation in Figure 2 (d) and (e). The effect of gradient thresholding ($th_g = 4$, in this case) is apparent in Figure 2 (f) and (g) in that it essentially eliminated the over-segmentation in the main background region although the character regions are still fragmented.

To identify the background region, we compute the sum of normalized intensity for all pixels of each region R_k:

$S_k = \sum_{(i,i)\in R_k} \left(y_{i,j} / 255 \right)$, and select the region with the largest S sum as the background region, where $y_{i,j}$ is the pixel intensity at location (i, j). For the example of Figure 2 (a), the segmentation results are shown in Figure 2 (h) and (i).

3. Image enhancements

Our enhancement components include illumination correction, color correction, and text sharpening.

3.1. Illumination correction

The illuminant surface may be estimated directly from the input image itself using the segmentation map. Assuming that the reflectivity of the document surface is uniform, illuminant values should be proportional to pixel luminance at background regions. Illuminant values at non-background regions may be interpolated from known background regions. One way to interpolate the two-dimensional surface is described in [5]. In practice, a separable 1-D (row and column) linear interpolation may be sufficient.

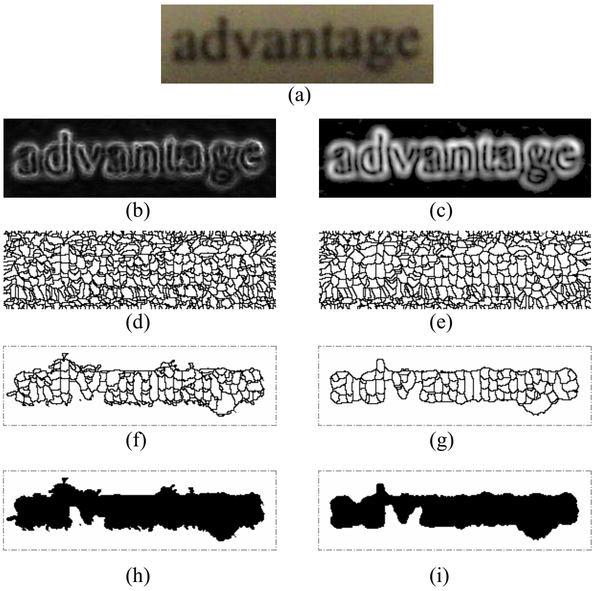


Figure 2. An example of background segmentation with the input image of (a). The left column shows the results of using a bilateral filter while the right column shows the results of a Gaussian filter. (b) and (c) are gradient images, (d) and (e) are watershed transforms, (f) and (g) are watershed transform using thresholded gradients, and (h) and (i) are segmentation results.

For an observed image $f(x, y)$ and the estimated illuminant surface $\hat{I}(x, y)$, the illumination-corrected image may be directly computed with

$$\hat{R}(x, y) = 255 * f(x, y) / \hat{I}(x, y).$$

3.2. Color correction

The segmentation is also used for color correction. Assuming that the true background color is a uniform neutral gray and that the observed average background color is $(\overline{R}_0, \overline{G}_0, \overline{B}_0)$, a set of three multipliers

$$(m_R, m_G, m_B) = \left(C_{\min} / \overline{R}_0, C_{\min} / \overline{G}_0, C_{\min} / \overline{B}_0 \right)$$

can be computed to convert the color $(\overline{R}_0, \overline{G}_0, \overline{B}_0)$ back into a neutral gray, where $C_{\min} = \min(\overline{R}_0, \overline{G}_0, \overline{B}_0)$. Notice that we assume the *R*,*G*,*B* color used here to be linear *R*,*G*,*B* values. The three multipliers are then applied to the *R*,*G*,*B* color planes of the whole image.

3.3. Selective sharpening

Unsharp masking is a simple and effective method for text sharpening. For a linear unsharp masking, the enhanced image may be expressed as

$$q(x,y) = (\beta + 1)\cdot p(x,y) - \beta \cdot g(x,y) \otimes p(x,y),$$

where $p(x,y)$ is an input image, $g(x,y)$ is a Gaussian lowpass kernel, β is a real number controlling the amount of sharpening, and $\otimes$ denotes a 2D convolution.

The unsharp masking may be applied to the illumination-corrected image. However, it is not desirable to apply the unsharp masking uniformly to all pixels since it may also amplify background noise. Instead, we selectively apply the unsharp masking only to non-background pixels. To take into account the sharpness of the input image, we adaptively determine the size of the Gaussian kernel and the amount of sharpening from the maximum gradient magnitude $g_{\max}$:

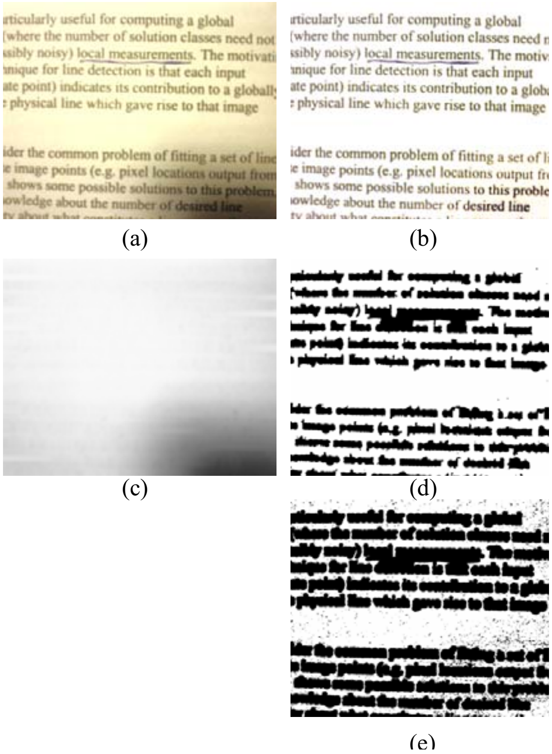
$$x = \begin{cases} x_{\min}, & if\,(g_{\max} > g_H) \\ [x_{\min}(g_{\max} - g_L) + x_{\max}(g_H - g_{\max})]/(g_H - g_L), & g_L \le g_{\max} \le g_H \\ x_{\max}, & if\,(g_{\max} < g_L) \end{cases}$$

where x is the parameter (window size or amount of sharpening) to be determined, $x_{\min}$ and $x_{\max}$ are the predetermined minimum and maximum values of the parameter, and g_L and g_H are the predetermined low and high reference gradient magnitude values.

4. Experimental results and discussion

In this section, we present experimental results of applying the proposed algorithm to several representative document images. The baseline parameters of the algorithm for the test images are the same: for pre-smoothing, 7×7 Gaussian kernel with $\sigma = 1.3$; for gradient thresholding, $th_{\min} = 4$ and $k = 0.5$; for selective unsharp masking, $g_H = 160$ and $g_L = g_H/3$, $w_{\min} = 3$ and $w_{\max} = 7$ for the window size, and $\beta_{\min} = 0.5$ and $\beta_{\max} = 3$ for the amount of sharpening.

Figure 3 (a) shows a poor-quality image with text-only content captured with a camera phone. Figure 3 (b) is the result with the proposed method. Figure 3 (d) is the segmentation result. Figure 3 (c) shows the estimated illuminant surface. As a comparison, we also show the segmentation result using a direct region growing in Figure 3 (e). In this case, we simply selected the largest connected component of pixels with gradient magnitude equal to zero (after thresholding). It can be seen clearly that the segmentation with direct region growing is significantly noisier and less accurate in identifying boundaries.


(a) (b)

(c) (d)

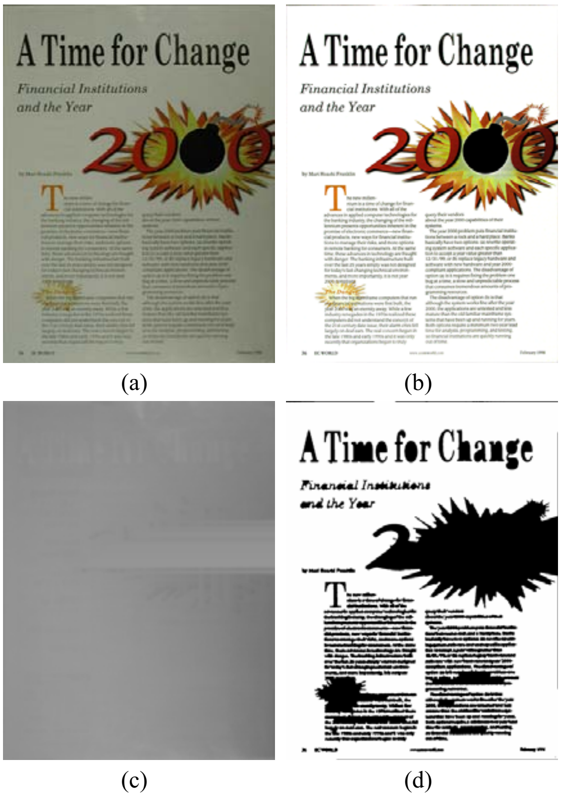
(e)

Figure 3. A text-only partial document. (a) the original image (1280×1024); (b) enhanced image with the proposed method; (c) estimated illumination surface; (d) segmentation map; (e) segmentation with direct region growing.

Figure 4 shows the case of a full-color magazine page. The original image (2048×1536, not shown) was captured using a 3MB Olympus C3000 digital camera. Figure 4 (a) is the image after rectification and cropping, and is the input for the enhancement algorithm. Figure 4 (b) is the result with the proposed method. Figure 4 (c) shows the estimated illuminant surface. Notice that the illumination and color correction worked well in making the background uniformly white without damaging the graphical regions.

Figure 5 shows a book page captured with a professional digital camera. The image is dominated by a very large picture region. Even though the background area is smaller, it was identified correctly and the enhancement result is satisfactory.

Figure 6 shows a whiteboard image with irregular hand drawings. Even though the segmentation algorithm missed some background areas enclosed by hand drawings, the illuminant surface is still estimated quite well and a good enhancement result is achieved.

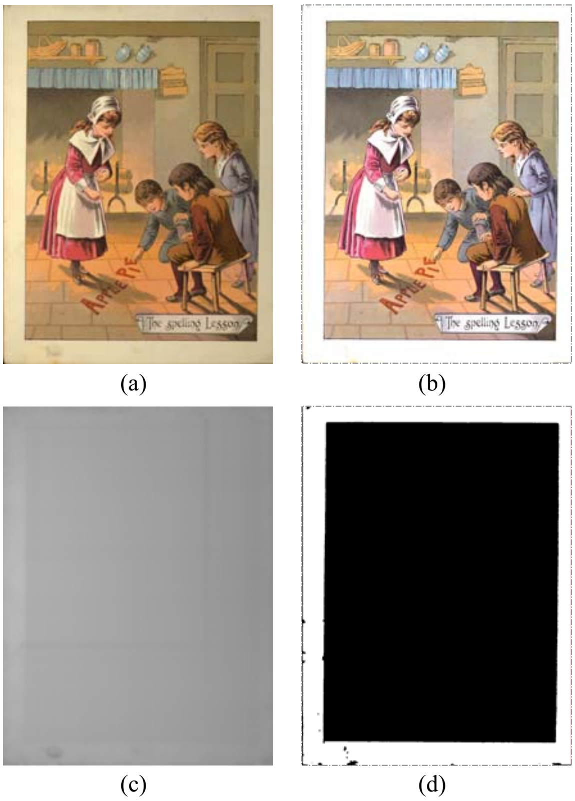


(a) (b)

(c) (d)

Figure 4. A full magazine page. (a) rectified and cropped image (1419×1878) for enhancement; (b) enhanced image with the proposed method; (c) estimated illumination surface; (d) segmentation map.

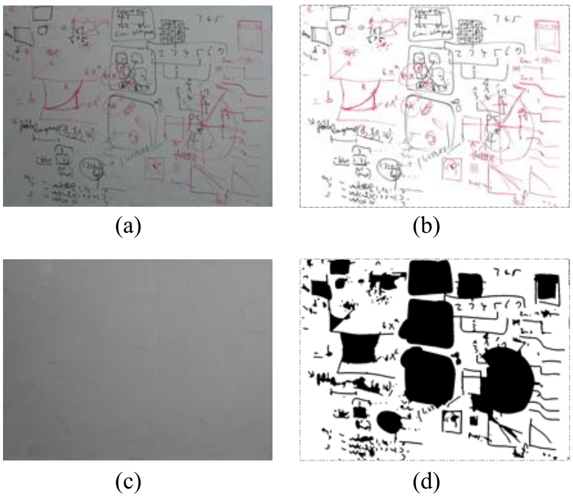
For the purpose of comparison, we implemented the block-based illumination correction algorithm proposed by *Hsia et al* [4]. The parameters for their algorithm include the size C of the sections of the raster scan-lines and the number M of maximum values for averaging. For the comparison results shown in Figures 7 and 8, two settings, with $M = 5$ and the number N of sections equal to 5 and 10, are included. The comparisons use the grayscale version of the two images, and only the illumination correction part of the proposed algorithm is applied. The results clearly verify theoretical analysis. For a large block size ($\mathrm{N} = 5$), the prominent, solid "explosion" figure is left largely intact. However, processing with the block size did not make the background uniform, as is evident on the right side of Figure 8 (c) and in the bottom right corner of Figure 7 (c). On the other hand, with smaller block size ($\mathrm{N} = 10$), background pixels are more uniformly white. However, damage to large objects (the "explosion" in Figure 8 (d) and handwritten underline in Figure 7 (d)) becomes evident. In contrast, illumination correction with the proposed method achieved satisfactory results in both cases.



(a) (b)

(c) (d)

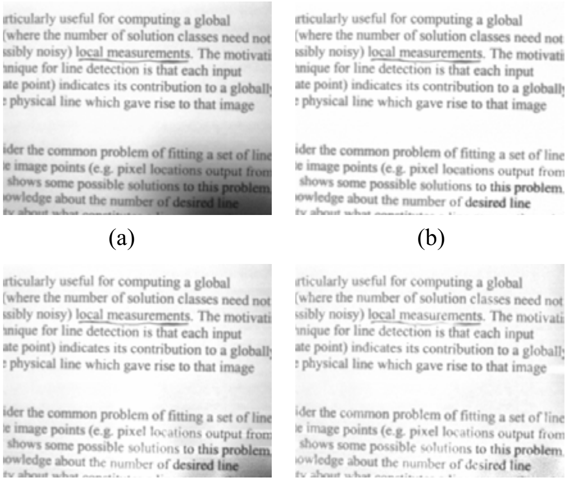
Figure 5. A figure-only book page. (a) cropped image (2409×3224) for enhancement; (b) enhanced image with the proposed method; (c) estimated illumination surface; (d) segmentation map.



(a) (b)

(c) (d)

Figure 6. A whiteboard image. (a) rectified and cropped image (1344×1002) for enhancement; (b) enhanced image with the proposed method; (c) estimated illumination surface; (d) segmentation map.



(a) (b)

(c) (d)

Figure 7. Comparison with Hsia *et al*'s method. (a) grayscale version of the image of Figure 3 (a). (b) illumination corrected using the proposed method; (c) correction with Hsia's method, with N = 5, (d) correction with Hsia's method, with N = 10.

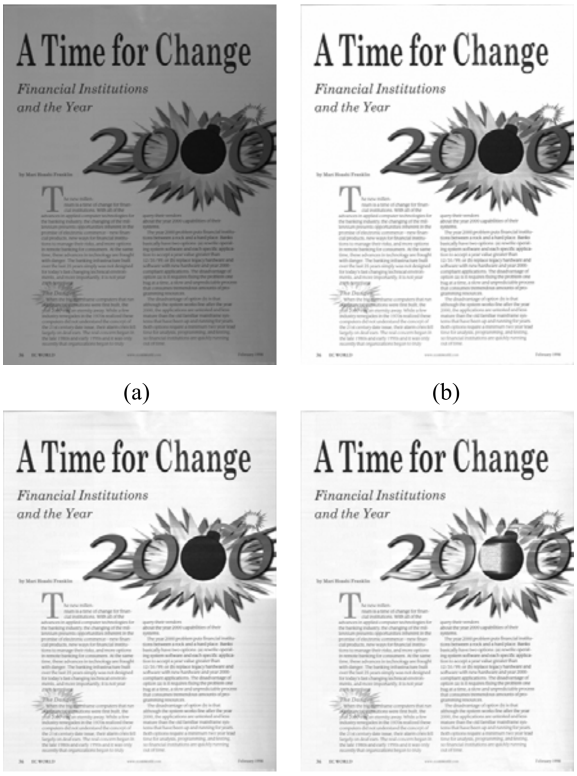
5. Conclusion

In this paper, we presented a segmentation-based image enhancement method and demonstrated its advantages over a fixed-block-based method. The segmentation is obtained using watershed transform on gradient magnitude. Both techniques lie on a solid mathematical foundation, and therefore their consistent behaviors can be expected. For the targeted enhancement task, we alleviate the over-segmentation problem by noise thresholding and by focusing on the background region. We demonstrated that satisfactory results can be achieved on images of various quality and content.

The main limitation of the current implementation lies in the assumption that the background region constitutes the largest catchment basin. Although this assumption is generally true for the large majority of documents, there are plenty of documents for which this assumption is invalid. In these cases, cross-region analysis is required to identify disconnected background regions. Other areas for future research include methods for better estimation of background noise and image sharpness.

6. References

[1] Jian Liang, David Doermann, Huiping Li, "Camera-based analysis of text and documents: a survey", IJDAR (2005) 7, p. 84–104



(a) (b)

(c) (d)

Figure 8. Comparison with Hsia *et al*'s method. (a) grayscale version of the image of Figure 4 (a). (b) illumination corrected using the proposed method; (c) correction with Hsia's method, with N = 5, (d) correction with Hsia's method, with N = 10.

[2] Rafael C. Gonzalez and Paul Wintz, *Digital image processing*, 2nd edition, Addison-Wesley, Reading, Massachusetts, 1987

[3] Pilu M., Pollard S., "A light-weight text image processing method for handheld embedded cameras", British Machine Vision Conference, Sept. 2002

[4] Shih-Chang Hsia, Ming-Huei Chen, and Yu-Min Chen, "A cost-effective line-based light-balance technique using adaptive processing", IEEE Trans. Image Proc., Vol. 15, No. 9, p. 2719-2729, Sept. 2006

[5] SD Yanowitz, AM Bruckstein, "A new method for image segmentation", CVGIP, v.46 n.1, p.82-95, April 1989

[6] Jos B.T.M. Roerdink and Arnold Meijster, "The watershed transform: definitions, algorithms and parallelization strategies", Fundamenta Informaticae 41 (2001) p. 187-228

[7] Vincent, L., and Soille, P. Watersheds in digital spaces: an e_cient algorithm based on immersion simulations. IEEE Trans. Patt. Anal. Mach. Intell. 13, 6 (1991), p. 583-598

[8] S. Beucher. The watershed transformation applied to image segmentation. Conference on Signal and Image Processing in Microscopy and Microanalysis, p. 299--314, September 1991

[9] Kongqiao Wang, Jari A. Kangas, Wenwen Li, “Character Segmentation of Color Images from Digital Camera”, ICDAR'01, p. 210-214

[10] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color Images”, ICCV, Bombay, India, 1998

[11] Danny Barash, “A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation”, IEEE Trans. PAMI, VOL. 24, NO. 6, p. 844-847, June 2002,

[11] D. Donoho. “De-noising by soft thresholding”, IEEE Trans. Information Theory, vol. 38(2), p. 613--627, 1995.

[12] Silvano Di Zenzo, “A note on the gradient of a multi-image”, CVGIP, Vol. 33, p. 116-125, 1986

[13] Jian Fan, “A local orientation coherency weighted color gradient for edge detection”, ICIP 05, p. 1132-1135, Sept. 2005

A Camera Based Digit Location and Recognition System for Garment Tracking

Anli Sun, Xi Wang, and Hong Wei
School of Systems Engineering, University of Reading, Reading, UK RG6 6AY
h.wei@reading.ac.uk

Abstract

Garment information tracking is required for clean room garment management. In this paper, we present a camera-based robust system with implementation of Optical Character Reconition (OCR) techniques to fulfill garment label recognition. In the system, a camera is used for image capturing; an adaptive thresholding algorithm is employed to generate binary images; Connected Component Labelling (CCL) is then adopted for object detection in the binary image as a part of finding the ROI (Region of Interest); Artificial Neural Networks (ANNs) with the BP (Back Propagation) learning algorithm are used for digit recognition; and finally the system is verified by a system database. The system has been tested. The results show that it is capable of coping with variance of lighting, digit twisting, background complexity, and font orientations. The system performance with association to the digit recognition rate has met the design requirement. It has achieved real-time and error-free garment information tracking during the testing.

1. Introduction

Automatic or semi-automatic *optical character recognition (OCR)* has been recognised as a successful technique and widely used in various applications for decades [1, 2]. Although OCR for typewritten text is considered as a solved problem generally, it is still challenging to develop a robust, efficient, and error free system, which not only deals with all noises and faded prints over time, but also has to distinguish between the target digits and other alphabetic characters in a same image in real-time manner. A unique system has to be designed and implemented to meet such requirements.

To the general purpose of OCR, many efforts have been made to solve problems of noise removal, text detection, and character recognition. Numerous algorithms have been developed to accomplish OCR with their own strengths and weakness, such as template matching, neural network, Gabor filters, wavelet based methods etc. [3, 4, 5]. Oliveira, *et. al.* developed a modular system using segmentation based verification strategies on automatic recognition of handwritten numerical strings for bank cheque authentication with an accuracy rate over 99% [2]. There are different types of noises or image defects (*i.e.* low contrast, blur, shading or out of focus) which affect performance of OCR as described in [3]. In order to improve robustness of a system to adapt wide variety of images, proper noise removal becomes important. Adaptive threshold algorithms can produce a global thresholding to convert color images with less complex background to binary images [5]. CCL has been used to detect text in images [4, 5, 6]. It may encounter difficulties with complex background where touching objects exist [7]. This requires additional contextual and structural features to be employed in the object search. ANNs have commonly been used as a recognition engine in OCR based systems [8]. Smagt compared three different neural networks on OCR and declared that ANNs with the BP learning algorithm had its capability of nonlinear projection and flexible network structure, and they were efficient and had a high recognition rate when the optimum number of layers and neurons was chosen against the number of characters to be recognised [9].

This paper presents a robust system for automatic digit location and recognition from real-time camera captured images as the purpose of clean room garment information tracking. The system employs a web camera as a sensor to capture images containing garment labels. The ROI consists of eight same font digits which are required to be recognised and recorded for garment tracking. In this paper, methodology used for image capturing, digit location, digit segmentation, recognition, and verification is discussed in Section 2. In Section 3, the experimental results are demonstrated. Section 4 concludes the paper and points out future work to optimise the system.

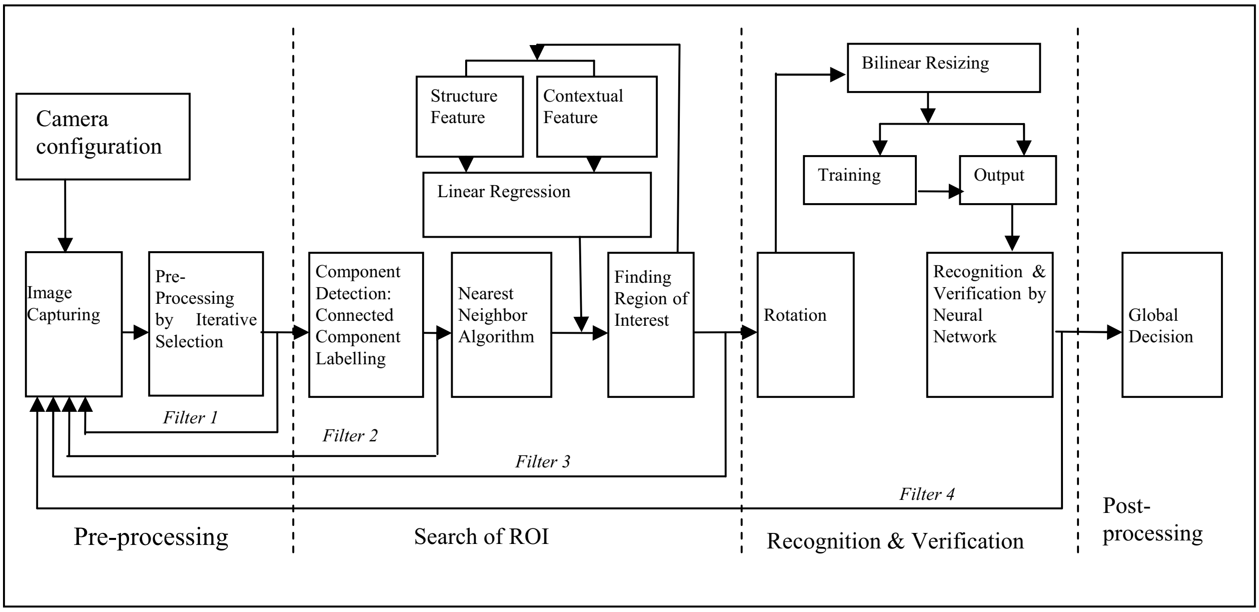


Figure 1. Schematic construction of the automatic digit location and recognition system

2. Methodology

The system layout is demonstrated in Figure 1. It consists of four parts: pre-processing, search of ROI, recognition and verification, and post-processing. In the pre-processing phase, a captured colour image is converted into a binary image using a proper thresholding value. Search of ROI is dedicated to processes of locating and segmentating the eight connected digits. The recognition and verification phase works with the neural network. The post-processing phase makes a decision on whether the processed digits should be accepted or rejected by the system and the database is accordingly updated. There are four filters involved in the system, which are used to control the capturing process. The details of the system are descibed as follows.

A. Image Capturing

The real-time image capturing process outputs images with 640x480 pixels in a speed of 30-50 fps (frames per second). It is controlled by a series of system filters (filtes 1, 2, 3 and 4). Filter 1 examines the overall grey level change between images with a timing interval to decide whether or not the captured image is a system targeted image, which should be passed to the next process stage. Filter 2 checks the number of objects produced from the CCL algorithm. If the number is not in a designated range (too many or too few objects are detected), the image under processing is abandoned, and new capturing is required. If the system does not find the ROI (eight connected digits), filter 3 will stop further process of the image and start the system from new images. When the system completes the recognition, a verification process is carried out with comparison of the recognised digits with garment labels stored in the database. If there is no match, filter 4 will trigger the image capturing process. The four filters play an important role in maintaining the system stability, and improving system efficiency and recognition accuracy.

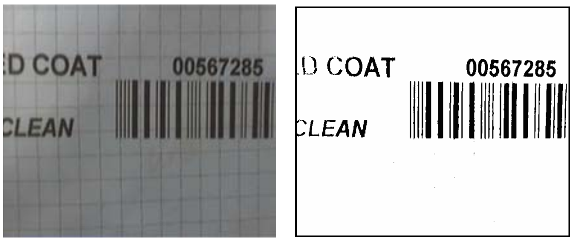
B. Image pre-processing to generate binary images

Image pre-processing is the essential step before the other algorithms can be applied in the further processing phases in the system. The purpose of the pre-processing is to convert colour images captured by the web-cam to binary images. Adaptively finding the best single threshold value is vital to effectively separate text prints and image background. There are a number of methods to find a threshold, such as using edge pixels, iterative selection, grey-level histograms, entropy, fuzzy sets, minimum error thresholding *etc.* [10]. Among them, an iterative selection algorithm is a refining process on an initial guess at a threshold by consecutively passing through the image. The system adapted this algorithm. Starting with the initial estimate of the threshold T_0, the *kth* estimate of the threshold can be written as

$$T_k = \frac{\sum_{i=0}^{T_{k-1}} i \cdot h[i]}{2\sum_{i=0}^{T_{k-1}} h[i]} + \frac{\sum_{j=T_{k+1}+1}^{N} j \cdot h[j]}{2\sum_{j=T_{k-1}+1}^{N} h[j]} \qquad (1)$$

where h is the histogram of the grey levels in the image, N is the maximum of greyscale value. When $T_k = T_{k+1}$, then T_k as the optimum threshold is found.

In practice, some garment labels have texture grids in the background and all printed texts are black. For example Figure 2(a) is a 640x480 colour image of garment barcode label captured by a web-cam. The iterative selection ensures that background grids were eliminated with the optimum threshold, as shown in Figure 2(b). In this case, the foreground and background was clearly separated with $T = 89$. Control of lighting conditions could be important to support the iterative selection algorithm in finding the optimum threshold.



(a) Original image **(b) After pre-processing**

Figure 2. Captured garment barcode label and its pre-processing result

Due to faded prints, lighting condition change, and inaccuracy of camera lens focus, the thresholding value generated by the iterative selection method could introduce problems of that the digits would be broken into small objects in confusing ROI search. To cope with such difficulties and increase system's robustness, the system takes the digit size as a feature to rule the final decision.

C. Detection of the ROI

As stated before, the ROI is 8 consecutive same font digits in an image outstanding of other prints (*e.g.* characters and barcode) in the same image. The process of ROI search operates at two steps: connected component labelling and object grouping based on the binary image generated in the pre-processing. Initially the CCL algorithm was used to search for all objects including all barcode, texts, digits, and other independent connected marks in the image**,** and then a nearest neighbour algorithm was applied to find the 8 nearest connected objects. There could be more than one object group with 8 consecutive objects in an image. New features of objects were used to decide the ROI. One of the features is that the expected ROI is an eight object group which is closest to the barcode with similar object size and similar gap between objects in the group. In this case, the position of barcode was used as a reference, and it has to be located firstly. The unique feature of these bars was identified in locating them in an image, *i.e.* they are a group of consecutive, highest and paralleled objects.

C.1. CCL algorithm

The CCL aims to label all pixels belonging to a same connected object. In the algorithm, a pixel P(x, y) can be labelled based on its 8 neighbours, which are defined as N8(P) ∊ {(x+1, y), (x-1, y), (x, y+1), (x, y+1), (x, y), (x+1, y+1), (x+1, y-1), (x-1, y+1) ,(x-1, y-1)}. The pixel 8-connectivity describes the relation among 8 neighbour pixels. Rules were followed to determine the pixel label based on a raster scan to the 8 neighbour pixels. At the end of processing, all pixels in the same object had a same label name. Figure 3 shows all the detected objects with colours in the image.

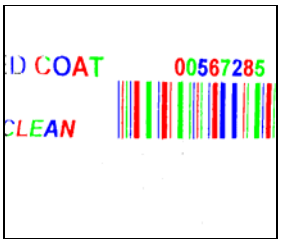


Figure 3. Different objects detected in the image

C.2. Nearest neighbour algorithm

The K-nearest-neighbour searching algorithm was employed to find the k nearest objects (in the case, the 8 consecutive digits) based on Euclidean distance from all the detected objects in the image [12]. The ROI was the combination of 8 nearest neighbour objects with similar size. If there were more than one object group with 8 consecutive objects, two further rules were applied to decide which group is the ROI. The first rule is that the Euclidean distance between the 8 objects must be similar; and the second rule exploited the barcode as a reference, *i.e.* the ROI has the nearest distance to the central line of the bars. Based on these rules, the ROI was located. Meanwhile, the 8 digits in the ROI were segmented and ready for recognition, as shown in Figure 4.

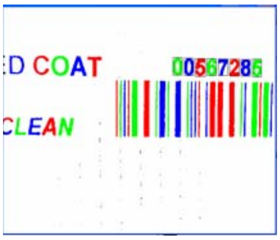


Figure 4. The ROI in the bounding boxes

C.3. Linear regression: a non-CCL method

The CCL based method can detect objects efficiently when the background is less complex. However it may fail when touching objects exist [13]. The background noises (*e.g.* embedded grids), image distortion (*e.g.* skew angles of fibres), lighting and shading condition, and faded prints cause the CCL based method failure in locating the ROI and segmenting digits. Under these circumstances, a complementary solution was used to reduce system errors and ensure that it is reliable and robust. Previous research has shown techniques to deal with such problems, for example a method treated texts as distinctive texture and used unsupervised clustering to classify each pixel as text or non text [14]. In our case, the special relationship (structural and contextual features) between the expected ROI and the barcode was taken into account. A simple geometrical method was designed, which is called the non-CCL method. A linear regression algorithm created a line through the centres of all bars. Moving the central line along the barcode in both dimensions, the gap between the digits and bars can be found as a unique feature to locate the ROI. The regression line also indicates the image orientation, with which the ROI can be rotated into the horizontal position. Figure 5 shows a line through the centres of bars. Image orientation was measured through the gradient of the line as 79.82 degree against the vertical coordinate axis in Figure 5. Eight digits were detected as shown in bounding boxes.

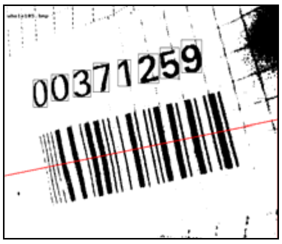


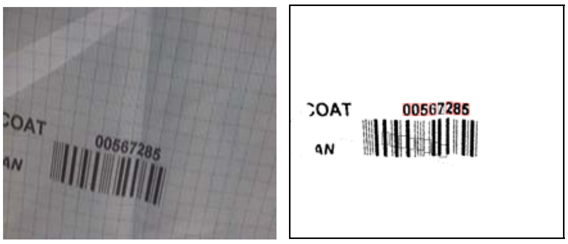
Figure 5. A line through centres of bars

D. Image rotation

The ROI found in images may have various orientations. Since the ANN recognition engine requires all digits in the horizontal to generate an array as inputs to the multilayer perceptron (MLP) neural network, the process of rotating the ROI is added into the system. The formula used is shown in Equation (2).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\partial & -\sin\partial \\ \sin\partial & \cos\partial \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (2)$$

where ∂ is the rotation angle when pixel $P(x,y)$ is rotated to $P(x', y')$ as the horizontal orientation. The example in Figure 6(a) is an original image captured by the camera. The orientation is random. After the rotation process, it is shown in Figure 6(b).



(a) Orginal Image **(b) after rotation**

Figure 6 Image rotation

With all eight digits are well segmented and refined, it is ready to be processed by the ANN recognition engine.

E. ANNs for digit recognition

ANNs have been applied for document analysis and recognition with a high recognition rate [15]. They process data by learning from training samples and have remarkable ability of coping with complicated and imprecise data with noises [16].

Our system adapted a one hidden layer fully connected MLP Neural Network with the BP learning algorithm. The MLP structure was configured with 6×8 input neurons and 10 output neurons which stand for 10 digits. To meet the input requirement for the MLP network, the bilinear interpolation algorithm was used to re-scale all digits in the size of 6x8 pixels to match the number of input neurons. The elements of resized 6×8 matrix were taken as 48 input neurons. The input values were bipolar (either 0s or 1s) which prepresent black and white in images, respectively. As one important factor for the whole training process, a training set which includes 10 groups of digits in various fonts was used. Among these training samples, digit twisting and distortion were taken into account.

Another vital parameter which could influence the training time and recognition accuracy is the number of hidden neurons in the hidden layer. Too many hidden neurons might exceed the optimal ANNs size due to the overfitting, which can lower the recognition ability, while too few hidden neurons may introduce large training errors [17]. Ideally, the least number of hidden neurons should be used, as it would be computational cost-effective, whilst still give the required performance. To find the optimum number, the network was trained and tested in turn by different numbers of hidden neurons, starting with five and adding another five each time.

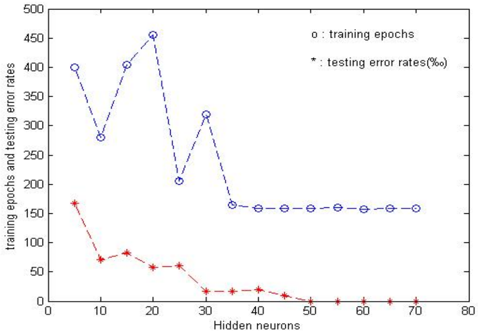


Figure 7. Epochs trend and testing errors against number of hidden neurons

The average training epoch values and testing error rates are demonstrated in Figure 7 based on 100 testing samples. It can be seen that circles which stand for training epochs converge to 160 when the number of hidden neurons was configured above 35. However, the error rate of recognition plotted by stars would not reduce to zero until 50 hidden neurons were used. Thus, to achieve both high recognition accuracy and fast training process, 50 hidden neurons were selected to construct the ANNs for the digit recognition purpose.

F. Verification

A verification mechanism was taken to ensure that the recognised digit sequence belongs to the unique counterpart in the system database. The system compares the recognition results with those strictly categorised in the database. If the database does not validate the recognised digit sequence, the system has to adjust its parameters to redo the whole process from the beginning until the digit sequence is accepted. Otherwise, an operator is involved in the process. It guarantees that the system is reliable and error free.

3. Experimental Evaluation

The system was designed to work in a commercial environment. It is not only required to be reliable, robust, and real-time, but more importantly is error-free. Therefore training and testing samples were carefully selected. Large amount of images captured from a variety of garments by a web camera (Logitech QuickCam Pro 4000, 1.3 Megapixel photo resolution, 640x480 digital video capture resolution) were used in testing. We present the testing results in three parts: detecting ROI (including detected by CCL and non-CCL), digit recognition, and verification. A variety of samples includes all orientations (0^{o}-360^{o}), different fonts, skewed images, embedded grids in the background, different colours of garments, lighting and shading, and faded prints. The recognition rate is defined as

$$\text{Recognition Rate} = \frac{\text{number of correctly recognised digits}}{\text{number of all testing digits}}$$

Table 1 shows that the detection of the ROI by the CCL was affected by sample skewing angles, embedded grids, lighting & shading, and faded prints. The non-CCL method was complementarily used in detecting the ROI where the CCL failed. The verification was performed to guarantee that all digit sequences output from the recognition system match those in the database. The error free for the system was achieved during the testing.

Table 1. System Performance Evaluation

Samples with variance of	No.	Detecting ROI		Recognition Rate %	Verified Rate %
		CCL %	Non-CCL %		
Orientations (0^{0}-360^{0})	450	100	0	100	100
Fonts	250	100	0	100	100
Skew Angles	150	84	16	100	100
Emb. Grids	150	64	36	97.4	100
Light.& Shad.	150	76	24	98.2	100
Faded Print	150	64	36	99.6	100

4. Conclusion

The camera-based automatic digit location and recognition system presented in this paper has been designed and developed specifically for the clean room garment management purpose. A web-camera is used for image capturing with consideration of cost-effective. The experimental tests have shown that the camera quality is satisfactory to the system usage. Various algorithms are adapted in the system for image pre-processing, object detection, segmentation, as well as digit recognition. The testing results have demonstrated the robustness of the system. Since the project is in its mid-stage, there are time spaces for further investigation of image pre-processing, object segmentation, *etc.*. More algorithms may be developed to complement the current algorithms used in the system to cope with more complex situations. Hardware support will also be taken into account in future development. Optical filters would be exploited to remove majority of visible lights to improve performance under light reflection caused by plastic covers of garment. Control of environmental lighting condition can be achieved by using an enclosed operating box with a fixed setup for a camera and lighting or more 'adaptive' thresholding algorithms.

5. Acknowledgement

The project is sponsored by the KTP/Micronclean Ltd. with project number of J3106800/01. The authors would like to thank Micronclean (Newbury, UK) Ltd. for the data supply.

References

[1] Victor Wu, R.Manmatha, and Edward M. Riseman, "Textfinder: an automatic system to detect and recognize text in images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 11, pp1224-1229, 1999

[2] Luiz s. Oliveira, Robert Sabourin, Flavio Bortolozzi, and Ching Y. Suen, "Automatic recognition of handwritten numerical strings: A recognition and verification strategy", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.24, No.11, pp1438-1454, 2002

[3] George Nagy, Thomas A. Nartkerb, and Stephen V. Ricec. "Optical character recognition: an illustrated guide to the frontier", *Proceedings of Document Recognition and Retrieval VII*, Vol.3967, pp58-69, San Jose, *SPIE*, January 2000

[4] O. Matan and C. Burges, "Recognizing overlapping hand-printed characters by centered-objects integrated segmentation and recognition", *Proc. Int'l Joint Conf. Neural Networks*, pp. 504-511, 1991.

[5] B. Yu and A. Jain. "A generic system for form dropout". *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 18, pp1127-1134, 1996

[6] Anil K. Jain and Bin Yu. "Automatic text location in images and video frames", *Pattern Recognition*, Vol.31, No.12, pp2055-2076, 1998

[7] Huiping Li, David Doermann and Omid Kia, "Automatic text detection and tracking", *IEEE Transaction on Image Processing*, Vol.9, No.1, pp147-156, 2000

[8] Jie Xi, Xian-Sheng Hua, Xiang-Rong Chen, Liu Wenyin, and Hong-Jiang Zhang, "A Video Text Detection and Recognition System", *Proc. of IEEE International Conference on Multimedia and Expo* (ICME 2001), August 22-25, Tokyo, Japan, pp1080-1083. 2001

[9] P. Patrick van der Smagt, "A comparative study of neural network algorithms applied to optical Character recognition", *Proc. of International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp1037-1044, 1990

[10] J. R. Parker, *Algorithms for Image Processing and Computer Vision*, John Wiley & Sons Inc, New York, 1996

[11] Rosenfeld, A., and Pfaltz, J. L. Sequential "Operations in digital picture processing", *J. ACM*, Vol.13, No.4, pp471-494, 1966

[12] Ting Liu, Andrew W. Moore, Alexander Gray, and Ke Yang. "An investigation of practical approximate nearest neighbour algorithms", *Proc. of Conference on Knowledge Discovery in Data*, pp 629-634, 2004

[13] Chiou-Shann Fuh, Shun-Wen Cho, and Kai Essig, "Hierarchical colour image region segmentation for content-based image retrieval system", *IEEE Transactions on Image Processing*, Vol. 9, No. 1, pp156-162, 2000

[14] A. K. Jain and S. Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing", *Mach. Vis. Appl.*, Vol.5, pp169-184, 1992

[15] S. Marini, M. Gori, and G. Soda, "Artificial neural networks for document analysis and recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol.27, No.1, pp23-35, 2005

[16] P. Picton, *Neural Networks, 2nd edition,* PALGRAVE, New York, 2000

[17] I. V. Tetko, D. J. Livingstone, and A. I. Luik, "Neural network studies 1. comparison of overfitting and overtraining", *J. Chem. Inf. Comput. Sci.,* Vol.35, pp826-833, 1995

Automatic Scene Text Recognition using a Convolutional Neural Network

Zohra Saidane and Christophe Garcia
Orange Labs
4, rue du Clos Courtel BP 91226
35512 Cesson Sévigné Cedex - France
firstname.lastname@orange-ftgroup.com

Abstract

This paper presents an automatic recognition method for color text characters extracted from scene images, which is robust to strong distortions, complex background, low resolution and non uniform lightning. Based on a specific architecture of convolutional neural networks, the proposed system automatically learns how to recognize characters without making any assumptions, without applying any pre-processing or post-processing and without using tunable parameters. For this purpose, we use a training set of scene text images extracted from the ICDAR 2003 public training database. The proposed method is compared to recent character recognition techniques for scene images based on the ICDAR 2003 public samples dataset in order to contribute to the state-of-the-art method comparison efforts initiated in ICDAR 2003. Experimental results show an encouraging average recognition rate of 84.53%, ranging from 93.47% for clear images to 67.86% for seriously distorted images.

1. Introduction

Natural text scene images contain important semantic information such as names of streets, institutes, shops, road signs, traffic information, etc.

While for printed document, optical character recognition (OCR) systems have already reached high recognition rates, and are widely commercialized, recognition of characters in natural scene images is still the subject of active research. In fact, this task is a challenging problem because of low resolution, complex background, non uniform lightning or blurring effects.

Most of the state of the art text image recognition methods are based on template matching which is also the case of most available commercial OCR systems mainly designed for printed text. This is an old principle that was proposed for OCR in 1929 by Tausheck. It reflects the technology at that time, which used optical and mechanical template matching. Light passed through mechanical masks is captured by a photo-detector and is scanned mechanically. When an exact match occurs, light fails to reach the detector and so the machine recognizes the characters printed on the paper. Nowadays, the idea of template matching is still used but with more sophisticated techniques. A database of models is created and matching is performed based on a distance measure. The models are generally composed of specific features that depend on the properties of the pattern to be recognized.

Chen at al [1] proposed a method based on character side profiles, in videos. First, a database is constructed with left, right, top and bottom side-profiles of sample characters. Then the candidate characters are recognized by matching their side-profiles against the database. This method requires of course a 'cleaned' binary text image. Therefore, they apply various pre-processing techniques before the recognition step, namely: shot boundary detection, edge-based text segmentation, multiple frame integration, gray-scale filtering, entropy-based thresholding, and noise removal using line adjacency graphs (LAGs). The authors reported a character recognition rate varying from 74.6% to 86.5% according to the video type (sport video, news, commercial videos).

Another template matching method was proposed by kopf et al. [3]. They have chosen to analyze the contour of a character and derive features extracted from the curvature scale space (CSS). This technique which is based on the idea of curve evolution, requires also binary text images. A CSS image is defined by the set of points where the curvature is null. In many cases, the peaks in the CSS image provide a robust and compact representation of a contour with concave segments. For characters without concave segments (e.g. 'I' and 'O'), the authors proposed the extended CSS method, where the original contour is mapped to a new contour with an inverted curvature, thus, convex segments become concave and the problem is solved.

The matching process is done by comparing the feature vectors (CSS peaks) of an unknown character to those of the characters that are stored in a database. It might be necessary to slightly rotate one of the CSS images to best align the peaks. In fact, shifting the CSS image left or right corresponds to a rotation of the original object in the image. Each character is stored only once in the database, and for instance, the horizontal moves compensate small rotations of italic character.

If a matching peak is found, the Euclidean distance of the height and position of each peak is calculated and added to the difference between the CSS images. Otherwise, the height of the peak in the first image is multiplied by a penalty factor and is added to the total difference. If a matching is not possible, the two objects are significantly different. This rejection helps to improve the overall results because noise or incorrectly segmented characters are rejected in the matching step. A recognition rate of 75.6% is reported for a test set of 2986 characters extracted from 20 artificial text images with complex background.

Yokobayashi et al [8, 9] proposed two systems for character recognition in natural scene images. Both of them rely on two steps: the first one is the binarization step and the second one is the recognition step based on an improved version of GAT correlation technique for grayscale character images.

In [8], the authors proposed a local binarization method applied to one of the Cyan/Magenta/Yellow color planes using the maximum breadth of histogram. This idea is based on the hypothesis that the more information entropy of grayscale occurrence a given image has the more effectively and easily a threshold value of binarization for the image can be determined, given that the breadth of grayscale histogram is closely related to the information entropy contained in the color plane. Therefore, they apply local binarization to the selected color plane. They compute the mean value of gray levels of all pixels in the selected color plane. Then, if a majority of nine pixels in a 3x3 local window have smaller gray levels than this mean, the pixel is considered as a character pixel, otherwise it is considered as a background pixel.

Once a binary image is obtained, an improved GAT correlation method [7] is applied for recognition. This is a matching measure between the binary character image and a template image. As templates, the authors use a single-font set of binary images of alphabets and numerals, which explains the need for the previously mentioned binarization step.

To obtain a measure robust to scale change, rotation, and possible distortion in general, the correlation should be computed on transformed images. Therefore, the authors search for optimal affine transformation components, which maximize the value of normalized cross-correlation, by using an objective function based on a Gaussian kernel. Once these parameters are determined, they compute the correlation between the transformed input image and a template image. Then, they compute the correlation between the input image and the transformed template image, and finally the average of these two values is used as the match score. The authors report an average recognition rate of 70.3%, ranging from 95.5% for clear images to 24.3% for little contrast images, from the ICDAR 2003 robust OCR sample dataset.

In [9], the authors proposed a binarization method based on three steps. Firstly, color vectors of all pixels in an input image are projected onto different arbitrarily chosen axis. Secondly, they calculate a maximum between-class separability by setting an optimal threshold according to the Otsu's binarization technique [6]. Thirdly, they select the axis that gives the largest between-class separability and the corresponding threshold for binarization of the input image. Then, they decide which class corresponds to characters or background according to the ratio of black pixels to white ones on the border of the binarized image. As in their previous work [8], once the binary image is obtained, an improved GAT correlation method is applied for recognition. The authors report an average recognition rate of 81.4%, ranging from 94.5% for clear images to 39.3% for seriously distorted images, from the ICDAR 2003 robust OCR sample dataset.

One can notice that in all the works mentioned above, there is a need for a pre-processing steps (i.e. binarization) and for finding optimal tunable parameters.

In this paper, we propose a novel automatic recognition scheme for natural color scene text images, based on supervised learning, without applying any pre-processing like binarization, without making any assumptions and without using tunable parameters. Moreover, our system makes a direct use of color information and insures robustness to noise, to complex backgrounds and to luminance variations.

The remainder of this paper is organized as follows. Section 2 describes in detail the architecture of the proposed neural network. It explains also the training process. Experimental results are reported in Section 3. Conclusions are drawn in Section 4.

2. The proposed recognition method

2.1. Architecture of the neural network

The proposed neural architecture is based on convolutional neural network architecture (CNN) [2, 4]. CNNs are hierarchical multilayered neural networks that combine three architectural ideas to ensure some degree of shift, scale, and distortion invariance:

- local receptive fields, to extract elementary features in the image

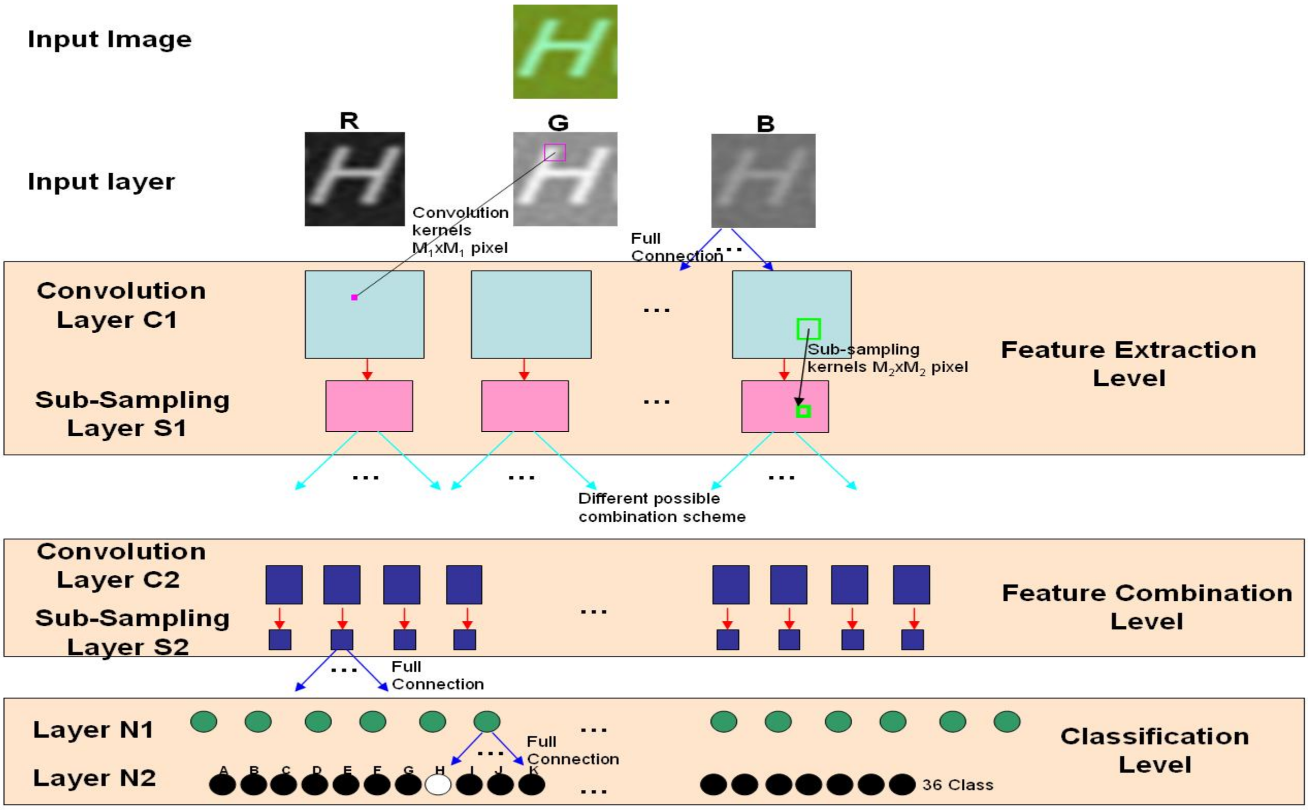


Figure 1. Architecture of the Network

- shared weights, to extract the same set of elementary features from the whole input image and to reduce the computational cost

- spatial sub-sampling, to reduce the resolution of the extracted feature maps.

As shown in Fig.1, the proposed network consists of seven different heterogeneous layers. Each layer contains feature maps which are the results of convolution, sub-sampling, or neuron unit operations. Applying and combining these automatically learnt operations ensure the extraction of robust features, leading to the automatic recognition of characters in natural images.

The first layer is the input layer E; it consists of $N_E = 3$ input maps, each of them corresponding to one color channel of the image, depending on the color space (RGB, YUV, CMY, etc.). Their pixel values are normalized to the range [-1, 1]. The RGB color space has been chosen in our experiments. We can distinguish three main levels:

LEVEL 1: Feature extraction level, relying on the C_1 and S_1 layers.

The layer C_1 extracts some specific features (oriented edges, corners, end points) directly from the three color channels. In fact, this layer is composed of NC_1 maps. Each unit in each map is connected to a $M_1 \times M_1$ neighborhood (biological local receptive field) in each of the color channels of the input layer. Furthermore, the trainable weights (convolutional mask) forming the receptive field, are forced to be equal for the entire map (weight sharing). A trainable bias is added to the results of each convolutional mask. Thus, each map can be considered as a feature map that has a learnt fixed feature extractor that corresponds to a pure convolution with a trainable mask, applied over the channels in the input layer. Multiple maps lead to the extraction of multiple features.

Once a feature is extracted its exact position is no longer important; only its relative position to other feature is relevant. Therefore, each map of the third layer S_1 results from local averaging and sub-sampling operations on a corresponding map in the previous layer C_1. So, the layer S_1 is composed of $NS_1 = NC_1$ feature maps. We use this sub-sampling layer to reduce by two the spatial resolution which reduces the sensitivity to shifts, distortions and variations in scale and rotation.

LEVEL 2: Feature combination level, relying on the C_2 and S_2 layers.

Layer C_2 allows extracting more complex information; outputs of different feature maps of layer S_1 are fused in order to combine different features. There are many possible combination schemes, the scheme used here will be explained later in section 2.3.

As in the first level, in this second level also, we consider a sub-sampling layer S_2, where each map results from local averaging and sub-sampling operations applied on a corresponding map in the previous layer C_2. Indeed, this progressive reduction of spatial resolution compensated by a progressive increase of the richness of the representation, which corresponds to the number of feature maps, enables a large degree of invariance to geometric transformations of the input.

LEVEL 3: Classification level, relying on the N_1 and N_2 layers. Each of them is a fully connected layer and contains classical neural units. The choice of the number of units in N_1 is empirical; whereas the number of units in N_2 depends on the number of classes to be recognized. If we consider the alphanumerical patterns, we will get 62 classes (26 lower case characters, 26 upper case characters and 10 digits). In order to reduce the number of output classes and consequently the number of neural weights to be learnt in between layers N_1 and N_2, we propose to use only 36 classes, where corresponding upper case and lower case characters are fused. This is made possible thanks to the strong generalization abilities of the proposed network.

2.2. The Database

We believe that using a public database is important to contribute to the clear understanding of the current state of the art. Therefore, we choose to use the ICDAR 2003 database, which can be downloaded from "http://algoval.essex.ac.uk/icdar/Datasets.html".

ICDAR 2003 proposed a competition named robust reading [5] to refer to text images that are beyond the capabilities of current commercial OCR packages. They chose to break down the robust reading problem into three sub-problems, and run competitions for each of them, and also a competition for the best overall system. The sub-problems are text locating, character recognition and word recognition. Due to the complexity of the database, contestants participated only to the text locating contest.

In this paper, we propose to perform character recognition on the ICDAR 2003 single character database. This database is divided into three subsets: a train subset (containing 6185 images), a test subset (containing 5430 images), and a sample subset (containing 854 images). These character images are of different size (5x12, 36x47, 206x223), different fonts, different colors, and present different kinds of distortion.

We used the train and the test subsets (a total of 11615 images) for training our network. Moreover, to enhance the robustness of the system, we increased the number of images in the training set by adding the corresponding negative images, and corresponding noisy images (we add Gaussian noise) of the original dataset to the final training set. Therefore, the final training set contains 34845 images.

We tested the performance of our system on the sample subset of 854 images.

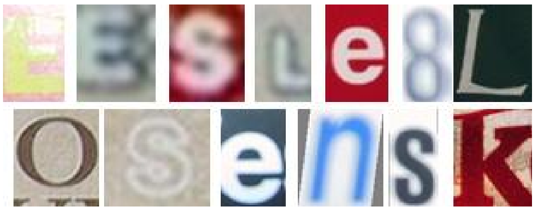


Figure 2. Examples of images of the training set

2.3. Training the network

We choose to normalize the size of all the images to 48 lines by 48 columns in RGB color space.

As mentioned before, we use 34845 color character scene images, $N_t = 30000$ in the training set and $N_v = 4845$ in the validation set. In fact, to avoid overfitting the training data and to increase the generalization ability of the system, a part of the whole training set is kept as validation set. This validation set is used to evaluate the network performance through iterations, as explained later on.

We choose to build $NC_1 = NS_1 = 6$ maps in layers C_1, and S_1; $NC_2 = NS_2 = 16$ maps in layers C_2, and S_2; 120 units in N_1; and $N_{Class} = 36$ units in layer N_2.

The combination scheme used is the following (figure 3): The first six C_2 feature maps take inputs from every contiguous subset of three feature maps in S_1. The next six take input from every contiguous subset of four feature maps in S_1. The next three take input from some discontinuous subsets of four feature maps in S_1. Finally, the last one takes input from all S_1 feature maps.

The convolution window size $M_1 \times M_1$=$M_2 \times M_2$=5×5 for both convolution layers C_1 and C_2. The sub-sampling factor is two in each direction for both sub-sampling layers S_1 and S_2.

We use linear activation functions in C_1 and C_2 and sigmoid activations fonctions in S_1, S_2, N_1 and N_2. The different parameters governing the proposed architecture, i.e.,

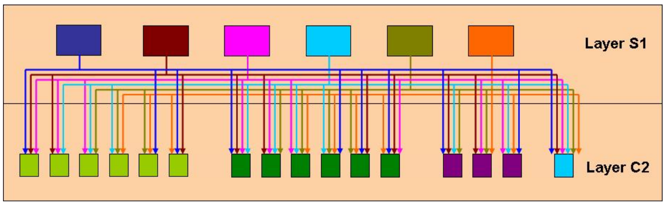


Figure 3. Feature Maps Combination Scheme

the number of layers, the number of maps, as well as the size of the receptive fields, have been experimentally chosen.

The training phase was performed using the classical back-propagation algorithm with momentum modified for being used in convolutional networks as described in [4] and it consists of the following steps:

1. Construct the desired output vector $\{D_h\}_{h=1..N_{Class}}$: for a given character image, belonging to class h, this desired output vector is constructed by setting its h^{th} element to 1, and setting the remaining elements to -1.

2. Compute the actual output vector $\{O_h\}_{h=1..N_{Class}}$: the character image is presented to the network as input, the last layer output vector represents the actual output vector.

3. Update weights: the weights are updated by backpropagation of the error between the actual output vector and the desired output vector.

4. Repeat step 1 until 3 for the N_t character images of the training set.

5. Compute the MSE (Mean Square Error) over the validation set: for every charater images of the validation set, repeat step 1 and 2, and then compute the MSE between the actual output vectors $\{O_{h,k}\}_{h=1..N_{Class},k=1..N_v}$ and the desired output vectors $\{D_{h,k}\}_{h=1..N_{Class},k=1..N_v}$ as follow:

$$MSE = \frac{1}{N_v \times N_{Class}} \sum_{k=1}^{N_v} \sum_{h=1}^{N_{Class}} (O_{h,k} - D_{h,k})^2 \tag{1}$$

6. Save the weights values if MSE is decreasing.

7. Repeat steps 1 until 6, until the desired number of iterations is reached.

After training, the system is ready to recognize automatically and rapidely color character images, without the need of any binarization preprocessing. In fact, the system is able to produce directly an output vector $\{O_h\}_{h=1..N_{Class}}$ for a given color input character image. The index corresponding to the highest component of this vector is considered as the recognized class.

After 40 training iterations, the proposed network achieves an average recognition rate of 91.77% on the whole training set.

3. Experimental results

To assess the performance of the proposed method, we use the sample subset of the ICDAR 2003 character database.

Table 1. Classification of images in ICDAR 2003 robust OCR Sample dataset.

Group	Number of Images
Clear	199
Background design	130
Multi-color character	54
Nonuniform lightning	40
Little contrast	37
Blurring	210
Serious distorsion	28
Total	698

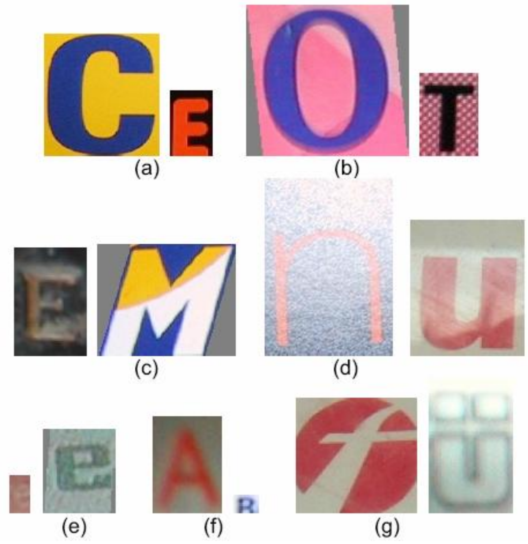


Figure 4. Examples of images in robust OCR Sample dataset classified into seven groups. (a) Clear. (b) Background design. (c) Multi-color character. (d) Nonuniform lightning. (e) Little contrast. (f) Blurring. (g) Shape distortion.

Given the wide range of variability contained in this database, and in order to compare our system to the recent works of Yokobayashi et al [8, 9], we consider the classification proposed in [8, 9] of 698 selected images from the above mentioned dataset, into seven groups according to the degree of image degradations and/or background complexity. Table 1 shows the number of images in each group and figure 4 shows examples of images in each of the seven groups.

Once training has been performed, the system is now ready to be used. We present the image to be recognized to the network after having resized it to the system retina size. Then we take the highest output of the network last layer and we consider the corresponding class as the recognized character class.

Processing the entire test set (698 images) takes about 26 seconds.

Figure 5 shows the results of our method compared to [8] and [9]. The performance of our system reaches a recognition rate of 84.53% which outperforms the methods [8] and [9]: it ranges from 67.86% for seriously distorted images to 93.47% for clear images. Compared to the methods [8] and [9], the performance of our system is less affected by the categorization of the test set, especially in the case of non-uniform lighting condition and serious distorsion, which is due to the good generalization ability of convolutional neural networks.

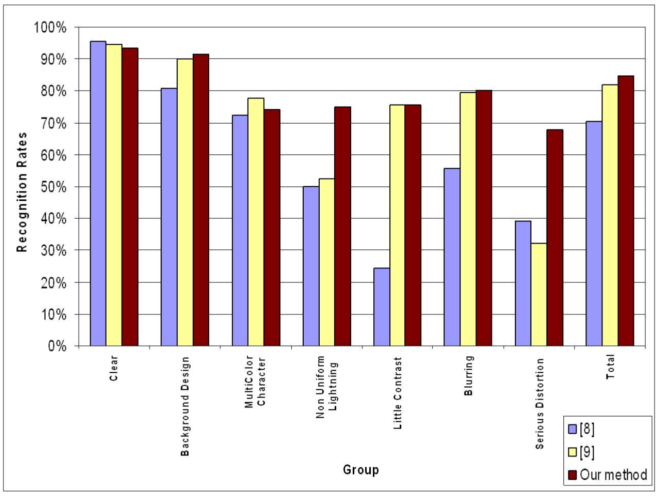


Figure 5. Recognition rates for each group of the test set

Figure 6 shows the the top N cumulative recognition rates, where the correct answer is within the N best answers (i.e. the N heighest outputs).

Here again our system outperforms the methods [8] and [9]. Furthermore, we notice that the cumulative recognition rate of the first two candidates is above 90%, showing the efficiency and the robustness of the proposed neural system.

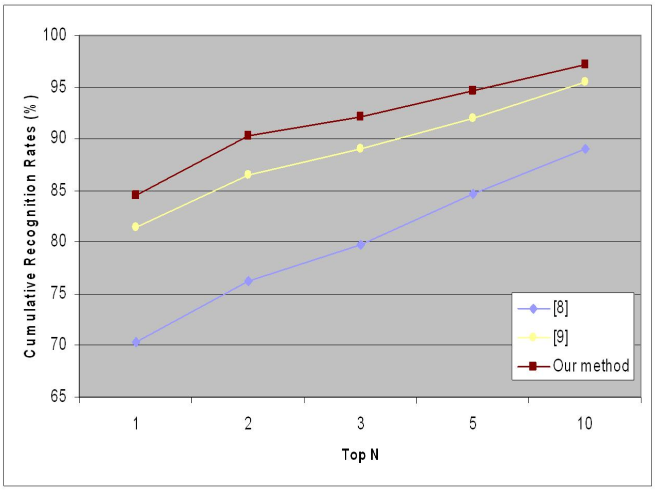


Figure 6. Cumulative recognition rates

4. Conclusion

In this paper, we have proposed an automatic recognition system for complex color scene text image recognition, based on a specific convolution neural network architecture. Thanks to supervised learning, our system does not require any tunable parameter and takes into account both color distributions and the geometrical properties of characters.

Only two state of the art methods have been tested on the public and actually challenging ICDAR 2003 robust reading data set. In this paper, we contribute to the state-of-the-art comparison efforts initiated in ICDAR 2003, and we show that our system outperforms these existing methods.

As future work, we plan to consider words recognition by including statistical language modeling in a more general network, using the proposed system has a building block.

References

[1]

[2] T. Chen, D. Ghosh, and S. Ranganath. Video-text extraction and recognition. *TENCON 2004, IEEE Region 10 Conference*, 1:319–322, Novembre 2004.

[3] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11), November 2004.

[4] S. Kopf, T. Haenselmann, and W. Effelsberg. Robust character recognition in low-resolution images and videos. Technical report, Department for Mathematics and Computer Science, University of Mannheim, April 2005.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. *Proc. of the IEEE*, November 1998.

[6] N. Otsu. A threshold selection method from gray-level histogram. *SMC-9*, 1979.

[7] T. Wakahara, Y. Kimura, and A. Tomono. Affine-invariant recognition of gray-scale characters using global affine transformation correlation. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-23:384–395, 20-24 Aug. 2001.

[8] M. Yokobayashi and T. Wakahara. Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. *Eighth International Conference on Document Analysis and Recognition ICDAR'05*, 1:167–171, 29 Aug.-1 Sept. 2005.

[9] M. Yokobayashi and T. Wakahara. Binarization and recognition of degraded characters using a maximum separability axis in color space and gat correlation. *18th International Conference on Pattern Recognition ICPR 2006*, 2:885–888, 20-24 Aug. 2006.

PhotoDoc: A Toolbox for Processing Document Images Acquired Using Portable Digital Cameras

Gabriel Pereira e Silva and Rafael Dueire Lins,
Departamento de Eletrônica e Sistemas – Universidade Federal de Pernambuco - Brazil
gfps@cin.ufpe.br, rdl@ufpe.br

Abstract

This paper introduces PhotoDoc a software toolbox designed to process document images acquired with portable digital cameras. PhotoDoc was developed as an ImageJ plug-in. It performs border removal, perspective and skew correction, and image binarization. PhotoDoc interfaces with Tesseract, an open source Optical Character Recognizer originally developed by HP and distributed by Google.

1. Introduction

Portable digital cameras are omnipresent in many ways of life today. They are not only an electronic device on their own right but have been embedded into many other devices such as portable phones and palmtops. Such availability has widened the range of applications, some of them originally unforeseen by their developers. One of such applications is using portable digital cameras to acquire images of documents as a practical and portable way to digitize documents saving time and the burden of having either to scan or photocopy documents. Figures 01 to 04 present different documents digitized using different camera models.

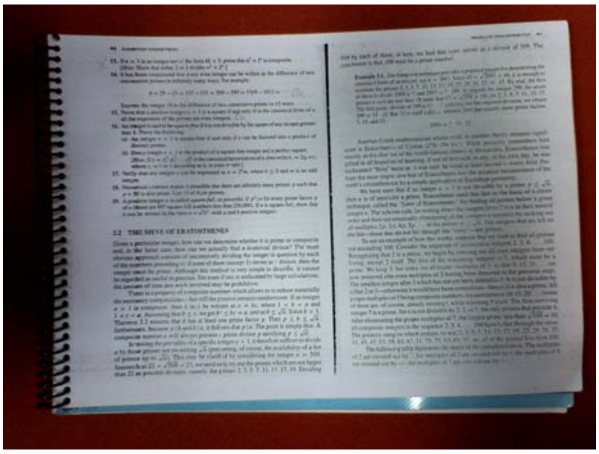


Figure 01 – Document image acquired with the camera of a LG cell phone KE-970 – (1600x1200 pixels) 409KB, without strobe flash

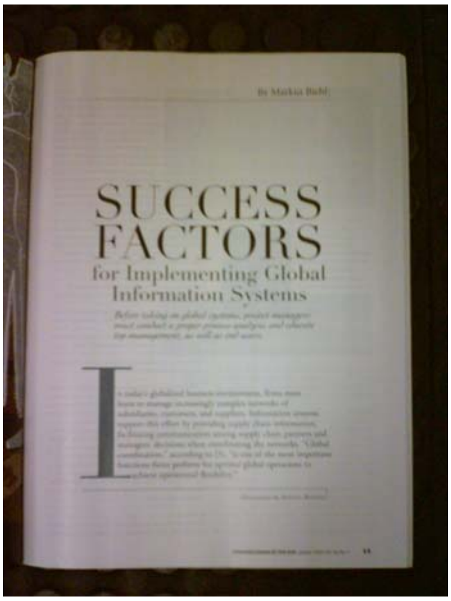


Figure 02 - Document image acquired with the camera of HP iPaq rx3700 (1280x960 pixels) 162KB, without strobe flash

This new use of portable digital cameras gave birth to a new research area [1][2] that is evolving fast in many different directions.

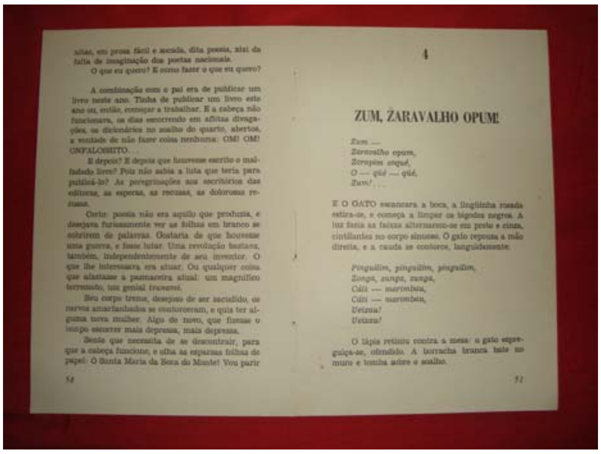


Figure 03 - Document image acquired with camera Sony DSC-P40 (4.1 Mpixels)

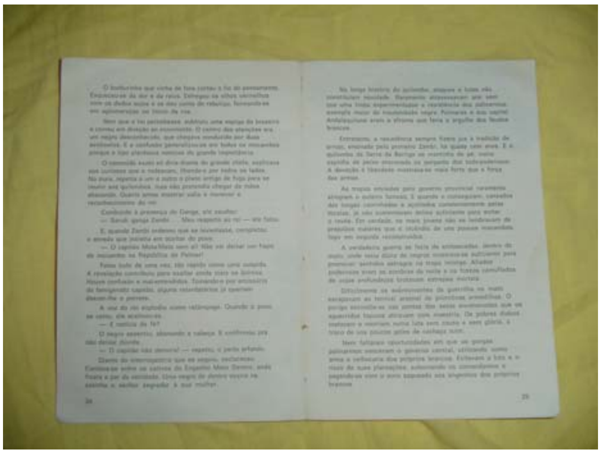


Figure 04 - Document image acquired with camera Sony DSC-P52 (3.2 Mpixels)

New algorithms, tools and processing environments are needed to provide users in general with simple ways of visualizing, printing, transcribing, compressing, storing and transmitting through networks such images. PhotoDoc is a processing environment developed to meet such needs.

The test documents used here were obtained in true-color, under different illumination conditions, with and without the inbuilt camera strobe flash, using a portable cell phone manufactured by LG KE-970 – (1600x1200 pixels) 409KB without strobe flash, a HP iPaq rx3700 (1280x960 pixels) 162KB without strobe flash, and two different models of portable cameras manufactured by Sony Corp. (models DSC-P52 and DSC-P40) of 3.2 and 4.1 Mega pixels, respectively. All cameras were set into "auto-focus" mode, i.e. the user leaves to the device the automatic setting of the focus.

Several specific problems arise in this digitalization process and must be addressed to provide a more readable document image, which also claims less toner to print, less storage space and consumes less bandwidth whenever transmitted through networks. The first of all is background removal as document photograph goes beyond the document size and incorporates parts of the area that surrounds it. The absence of mechanical support for taking the photo yields a non-frontal perspective that distorts and skews the document image. The distortion of the lenses of the cameras makes the perspective distortion not being a straight but a convex line, depending on the quality of the lens and the relative position of the camera and the document. Non-uniform illumination of the environment and strobe flash, whenever available in the device adds difficulties in image enhancement and binarization.

2. The PhotoDoc Environment

PhotoDoc was conceived as a device independent software tool to run on PCs. Whenever the user unloads his photos he will be able to run the tool prior to storing, printing or sending through networks the document images. The algorithms and the basic functionality of PhotoDoc may be incorporated to run on a device such as a PDA or even a camera itself. Such a possibility is not considered further herein. Due to implementation simplicity and portability the current version of PhotoDoc was implemented as an ImageJ [20] Plug-in. ImageJ is an open source image processing environment in Java developed by Wayne Rasband, is at the Research Services Branch, National Institute of Mental Health, Bethesda, Maryland, USA. Figure 05 shows a screen shot of PhotoDoc being activated from ImageJ.

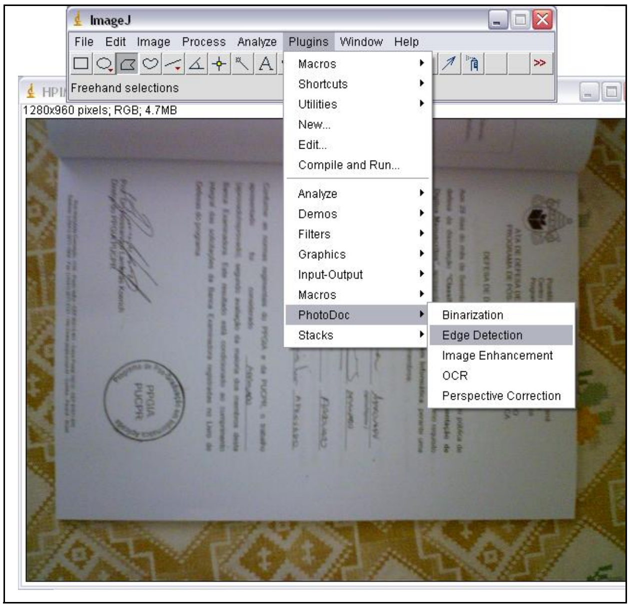


Figure 05 – PhotoDoc plugin in ImageJ

As one may observe in Figure 05, the present version of the PhotoDoc plug-in offers five different filters, which appear in alphabetical order:

- Binarization
- Edge Detection
- Image Enhancement
- OCR, and
- Perspective Correction and Crop

The fact that PhotoDoc is now in ImageJ also allows the user to experiment with the different filters and other plug-ins already present in ImageJ. It is important to stress, however, that ImageJ as an open code library allows the developer to extract from it only the needed functionality in such a way that the developer may provide to ordinary user a PhotoDoc interface that looks independent from ImageJ. At present, the authors of this paper consider such possibility premature. Such tool particularization seems to be more adequate is coupled with a particular device, which allows also a better fine tuning of the algorithms developed for and implemented in PhotoDoc. In what follows the PhotoDoc filter operations are described.

3. A New Border Detection Algorithm

The very first step to perform in processing a document image in PhotoDoc is to detect the actual physical limits of the original document [3]. The algorithm presented in reference [7] was developed based on images acquired by 3 and 4 Mpixel cameras. Unfortunately, its performance in lower resolution cameras has shown to be inadequate.

A new edge detection algorithm, based on ImageJ filters, was developed and is presented herein. This new algorithm behaved suitably on a wide range of images with different kinds of paper, including glossy ones. The new algorithm was obtained by composing existing filters in ImageJ. The steps of the new border removal algorithm are:

1. Process + Enhance Contrast.
2. Process + Find Edges.
3. Image Type 8 bits.
4. Image Adjust Threshold – Black and White value 122.
5. Process Binary + Erode.

At this point the resulting image provides well defined borders that allow finding the document edges. Figure 06 presents the result of applying the steps of the algorithm above to the document image presented in Figure 01, which appears in the top-left corner, until reaching the resulting image in the bottom-right one.

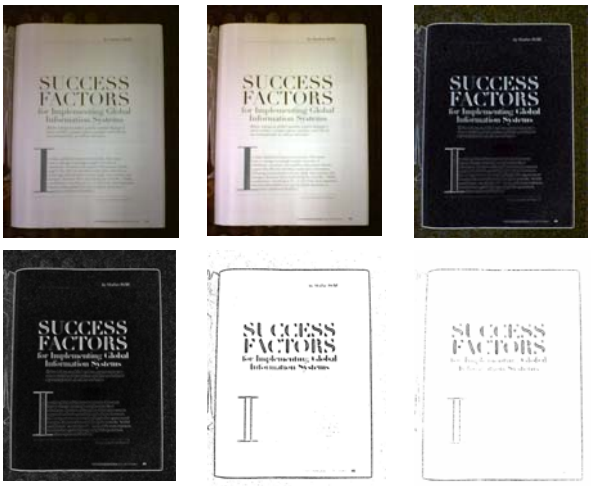


Figure 06 – Step by step filtering of image for border edge detection using ImageJ

The result of the activation of button "Edge Detection" on a document in PhotoDoc yields an image such as the one presented on Figure 07.

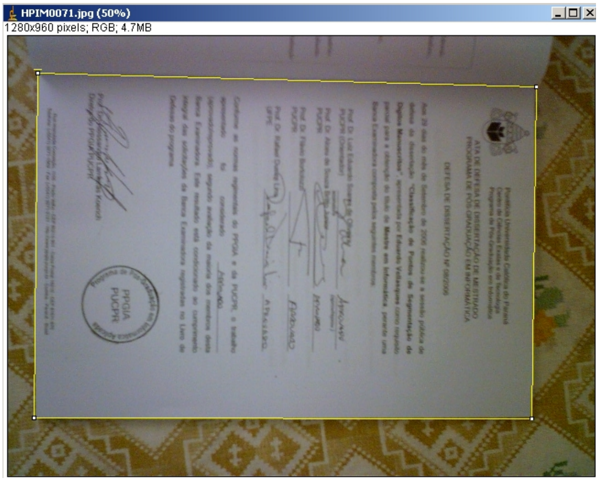


Figure 07 - Document image with edges (in yellow) automatically detected by PhotoDoc

Although the algorithm presented correctly detected edges for all the tested documents, the Edge Detection filter in PhotoDoc allows the user to adjust the edges by dragging along the four corners of the document image. This may also be of help whenever the document photo is not completely surrounded by a border of whenever strong uneven illumination causes edges not to be detected.

4. Perspective Correction and Crop

PhotoDoc incorporates a filter to correct the distortion and skew of the image introduced by the non-frontal position of the camera in relation to the document. A pinhole model for the camera was adopted [5, 6, 9] and provides a simple way to solve the problem at first. The experiments reported in [8] point at, narrowing edges and using bi-cubic interpolation as the rule-of-thumb to yield images more pleasant for the human reader and also with less transcription errors in OCR response. The difficulty inherent to such transformation is finding the four corners of the original image that will be taken as the basis for the correction. Edge or border detection, as explained above, is the first step the image should undergo. Once the document edges are determined as shown in Figure 07 the "Perspective Correction" filter in PhotoDoc may be called as shown in Figure 08.

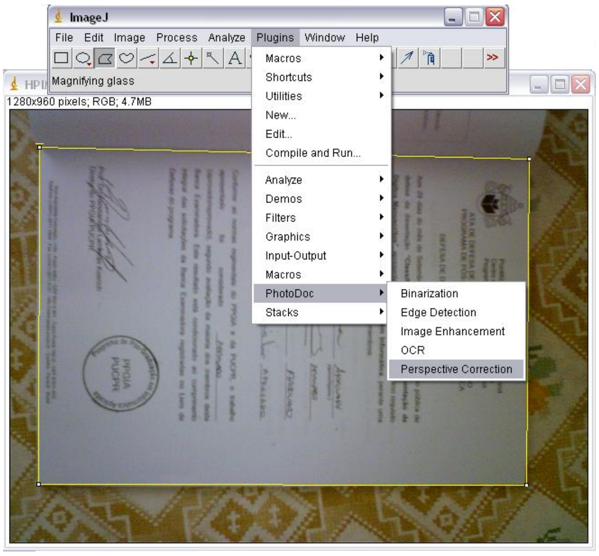
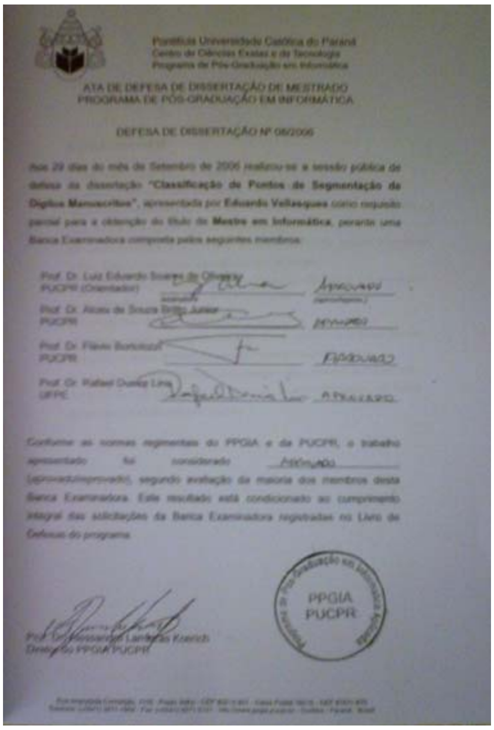


Figure 08 – Activating Perspective Correction in edge detected image from Figure 07 in PhotoDoc

PhotoDoc will automatically perform a perspective, skew, and orientation correction and then crop the resulting image, yielding a document as the one shown in Figure 09. One should observe that the cropped document whenever visualized in a screen display provides a far better image to the human reader, if printed saves toner and yields a more readable document, and whenever stored or transmitted through computer networks saves on average 25% of space [7].



Pontifícia Universidade Católica do Paraná
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Informática

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DEFESA DE DISSERTAÇÃO Nº 08/2006

Aos 29 dias do mês de Setembro de 2006 realizou-se a sessão pública de defesa da dissertação **"Classificação de Pontos de Segmentação de Dígitos Manuscritos"**, apresentada por **Eduardo Vellasques** como requisito parcial para a obtenção do título de **Mestre em Informática**, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Luiz Eduardo Soares de Oliveira
PUCPR (Orientador) Aprovado
(aprov/reprov.)

Prof. Dr. Alceu de Souza Britto Junior
PUCPR Aprovado

Prof. Dr. Flávio Bortolozzi
PUCPR Aprovado

Prof. Dr. Rafael Dueire Lins
UFPE Aprovado

Conforme as normas regimentais do PPGIA e da PUCPR, o trabalho apresentado foi considerado Aprovado (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora. Este resultado está condicionado ao cumprimento integral das solicitações da Banca Examinadora registradas no Livro de Defesas do programa.

Prof. Dr. Alessandro Lameiras Koerich
Diretor do PPGIA PUCPR

PPGIA
PUCPR

Figure 09 – Cropped document after perspective, skew and orientation correction by PhotoDoc

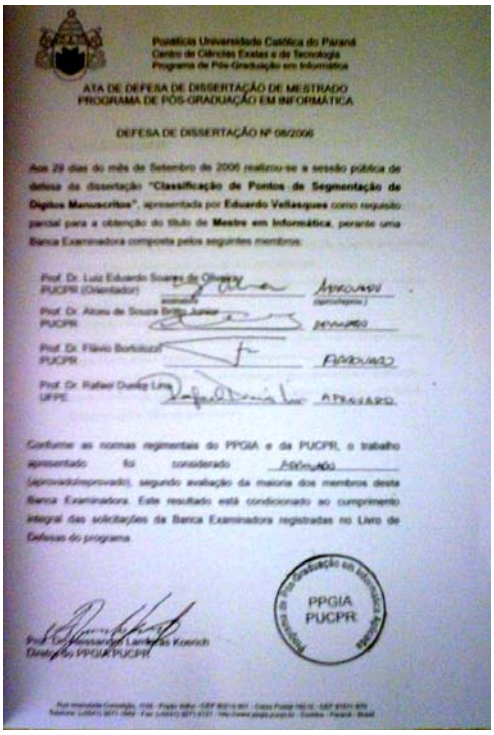
The perspective correction algorithm in PhotoDoc was implemented using the JAI (Java Advanced Imaging) library [23].

5. Image Enhancement

There is a paramount number of possibilities to improve the quality of document images depending of the features of the camera, such as its resolution, lens distortion, the quality of embedded algorithms, environment illumination, quality and color of the paper, etc. Devices have many different technical characteristics, thus it is impossible to find a general solution that would suit all of them, overall with their fast technological evolution pace. However, in the case of devices with embedded flash, if it was not used the resulting document photograph "looked" slightly blurred (out-of-focus). Most possibly, this is related with the fact that the diaphragm of the objective stayed open for much longer to compensate the loss in illumination. As no mechanical support was used to stabilize the camera, chances are that the photographer moved briefly during the shot. Some other times, a slight inclination of the camera in a clear environment may be enough to the luminosity sensor to assume that there is no need for the camera to provide extra illumination, canceling the flash activation. Thus, in order to minimize these factors one may recommend that document photos are:

1. Taken with the embedded flash of the camera set as "on", forcing its activation regardless of the luminosity of the environment.
2. Obtained indoors.

Several of the ImageJ image enhancement algorithms were tested. Non-uniform illumination brings a high degree of difficulty to the problem. A general algorithm that provided gains in all the images studied weakening the illumination problem was provided by the "Enhance Contrast" filter in ImageJ, as may be observed in the image presented in Figure 10.



Pontifícia Universidade Católica do Paraná
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Informática

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DEFESA DE DISSERTAÇÃO Nº 08/2006

Aos 29 dias do mês de Setembro de 2006 realizou-se a sessão pública de defesa da dissertação **"Classificação de Pontos de Segmentação de Dígitos Manuscritos"**, apresentada por **Eduardo Vellasques** como requisito parcial para a obtenção do título de **Mestre em Informática**, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Luiz Eduardo Soares de Oliveira
PUCPR (Orientador) ______ Aprovado
assinatura (aprovado/reprovado)

Prof. Dr. Alceu de Souza Britto Junior
PUCPR ______ Aprovado

Prof. Dr. Flávio Bortolozzi
PUCPR ______ Aprovado

Prof. Dr. Rafael Dueire Lins
UFPE ______ Aprovado

Conforme as normas regimentais do PPGIA e da PUCPR, o trabalho apresentado foi considerado Aprovado (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora. Este resultado está condicionado ao cumprimento integral das solicitações da Banca Examinadora registradas no Livro de Defesas do programa.

Prof. Dr. Alessandro Lameiras Koerich
Diretor do PPGIA/PUCPR

Programa de Pós-Graduação em Informática Aplicada
PPGIA
PUCPR

[illegible]
[illegible]

Figure 10 – PhotoDoc enhanced version of Figure 09.

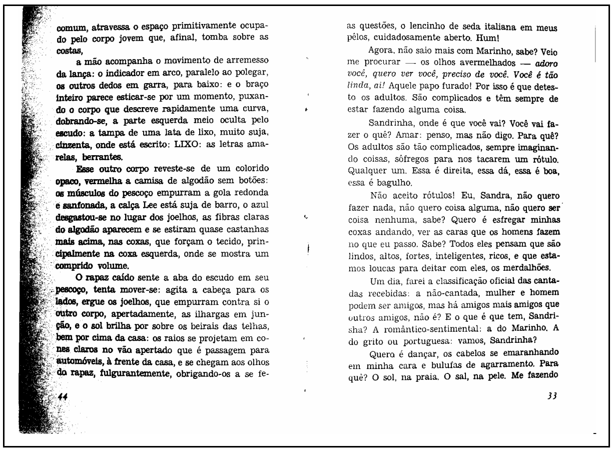
The "Enhance Contrast" filter in ImageJ performs histogram stretching. The *Saturated Pixels* value determines the number of pixels in the image that are allowed to become saturated. Increasing this value will increase contrast. This value should be greater than zero to prevent a few outlying pixels from causing the histogram stretch to not work as intended. For the case of PhotoDoc the best results were obtained with 0.5% of saturated pixels.

6. Document Binarization

Monochromatic images are the alternative of choice for most documents with no iconographic or artistic value saving storage space and bandwidth in network transmission. Most OCR tools pre-process their input images into grayscale or binary before character recognition. Reference [8] reports on the binarization of documents acquired with portable digital cameras. Fifty test images obtained with 3.2 and 4.1 Mpixel cameras (Sony DSC-P52 and DSC-P40, respectively) had their borders removed and were perspective and skew corrected before binarization, both globally and also splitting the images into 3, 6, 9 and 18 regions [16]. The following algorithms were tested:

1. Algorithm 1 – da Silva *et al.* [10];
2. Algorithm 2 – Mello *et al* [10];
3. Algorithm 3 – Pun [11];
4. Algorithm 4 – Kapur-Sahoo-Wong [12];
5. Algorithm 5 – Wu-Songde-Hanqing [13];
6. Algorithm 6 – Otsu [14];
7. Algorithm 7 – Yen-Chang-Chang [15];
8. Algorithm 8 – Johannsen-Bille [16].

According to [8], the global algorithm that yielded the best results both for visual inspection and in OCR response was the entropy based algorithm by da_Silva *et al* [10] (Figure 11). The best results obtained by applying the binarization algorithms in regions of the documents were provided by the algorithm by Kapur-Sahoo-Wong [12] with 18 regions (Figure 12).



comum, atravessa o espaço primitivamente ocupado pelo corpo jovem que, afinal, tomba sobre as costas,

a mão acompanha o movimento de arremesso da lança: o indicador em arco, paralelo ao polegar, os outros dedos em garra, para baixo: e o braço inteiro parece esticar-se por um momento, puxando o corpo que descreve rapidamente uma curva, dobrando-se, a parte esquerda meio oculta pelo escudo: a tampa de uma lata de lixo, muito suja, cinzenta, onde está escrito: LIXO: as letras amarelas, berrantes.

Esse outro corpo reveste-se de um colorido opaco, vermelha a camisa de algodão sem botões: os músculos do pescoço empurram a gola redonda e sanfonada, a calça Lee está suja de barro, o azul desgastou-se no lugar dos joelhos, as fibras claras do algodão aparecem e se estiram quase castanhas mais acima, nas coxas, que forçam o tecido, principalmente na coxa esquerda, onde se mostra um comprido volume.

O rapaz caído sente a aba do escudo em seu pescoço, tenta mover-se: agita a cabeça para os lados, ergue os joelhos, que empurram contra si o outro corpo, apertadamente, as ilhargas em junção, e o sol brilha por sobre os beirais das telhas, bem por cima da casa: os raios se projetam em cones claros no vão apertado que é passagem para automóveis, à frente da casa, e se chegam aos olhos do rapaz, fulgurantemente, obrigando-os a se fe-

44

as questões, o lencinho de seda italiana em meus pêlos, cuidadosamente aberto. Hum!

Agora, não saio mais com Marinho, sabe? Veio me procurar — os olhos avermelhados — *adoro você, quero ver você, preciso de você. Você é tão linda, ai!* Aquele papo furado! Por isso é que detesto os adultos. São complicados e têm sempre de estar fazendo alguma coisa.

Sandrinha, onde é que você vai? Você vai fazer o quê? Amar: penso, mas não digo. Para quê? Os adultos são tão complicados, sempre imaginando coisas, sôfregos para nos tacarem um rótulo. Qualquer um. Essa é direita, essa dá, essa é boa, essa é bagulho.

Não aceito rótulos! Eu, Sandra, não quero fazer nada, não quero coisa alguma, não quero ser coisa nenhuma, sabe? Quero é esfregar minhas coxas andando, ver as caras que os homens fazem no que eu passo. Sabe? Todos eles pensam que são lindos, altos, fortes, inteligentes, ricos, e que estamos loucas para deitar com eles, os merdalhões.

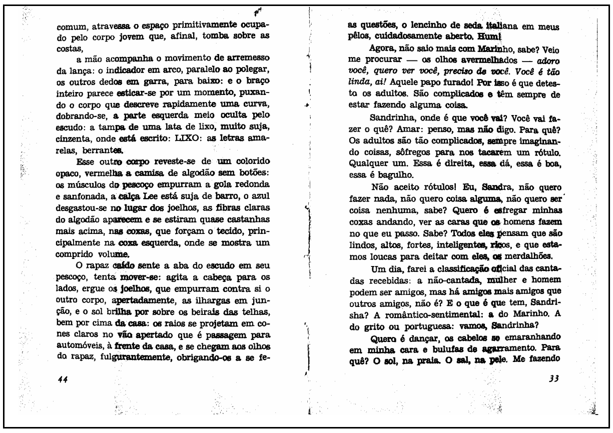
Um dia, farei a classificação oficial das cantadas recebidas: a não-cantada, mulher e homem podem ser amigos, mas há amigos mais amigos que outros amigos, não é? E o que é que tem, Sandrisha? A romântico-sentimental: a do Marinho. A do grito ou portuguesa: vamos, Sandrinha?

Quero é dançar, os cabelos se emaranhando em minha cara e bulufas de agarramento. Para quê? O sol, na praia. O sal, na pele. Me fazendo

33

Figure 11. Binarized with da Silva *et al.*(global)

Some new algorithms were tested herein including Sauvola [18], Niblack [17] and MROtsu [14]. The results obtained may be found in Figures 13 to 15. Unfortunately, the results obtained for binarizing images captured by the devices without embedded strobe flash were unsatisfactory. Further analysis and pre-processing must be studied for such images.



comum, atravessa o espaço primitivamente ocupado pelo corpo jovem que, afinal, tomba sobre as costas,

a mão acompanha o movimento de arremesso da lança: o indicador em arco, paralelo ao polegar, os outros dedos em garra, para baixo: e o braço inteiro parece esticar-se por um momento, puxando o corpo que descreve rapidamente uma curva, dobrando-se, a parte esquerda meio oculta pelo escudo: a tampa de uma lata de lixo, muito suja, cinzenta, onde está escrito: LIXO: as letras amarelas, berrantes.

Esse outro corpo reveste-se de um colorido opaco, vermelha a camisa de algodão sem botões: os músculos do pescoço empurram a gola redonda e sanfonada, a calça Lee está suja de barro, o azul desgastou-se no lugar dos joelhos, as fibras claras do algodão aparecem e se estiram quase castanhas mais acima, nas coxas, que forçam o tecido, principalmente na coxa esquerda, onde se mostra um comprido volume.

O rapaz caído sente a aba do escudo em seu pescoço, tenta mover-se: agita a cabeça para os lados, ergue os joelhos, que empurram contra si o outro corpo, apertadamente, as ilhargas em junção, e o sol brilha por sobre os beirais das telhas, bem por cima da casa: os raios se projetam em cones claros no vão apertado que é passagem para automóveis, à frente da casa, e se chegam aos olhos do rapaz, fulgurantemente, obrigando-os a se fe-

44

as questões, o lencinho de seda italiana em meus pêlos, cuidadosamente aberto. Hum!

Agora, não saio mais com Marinho, sabe? Veio me procurar — os olhos avermelhados — *adoro você, quero ver você, preciso de você. Você é tão linda, ai!* Aquele papo furado! Por isso é que detesto os adultos. São complicados e têm sempre de estar fazendo alguma coisa.

Sandrinha, onde é que você vai? Você vai fazer o quê? Amar: penso, mas não digo. Para quê? Os adultos são tão complicados, sempre imaginando coisas, sôfregos para nos tacarem um rótulo. Qualquer um. Essa é direita, essa dá, essa é boa, essa é bagulho.

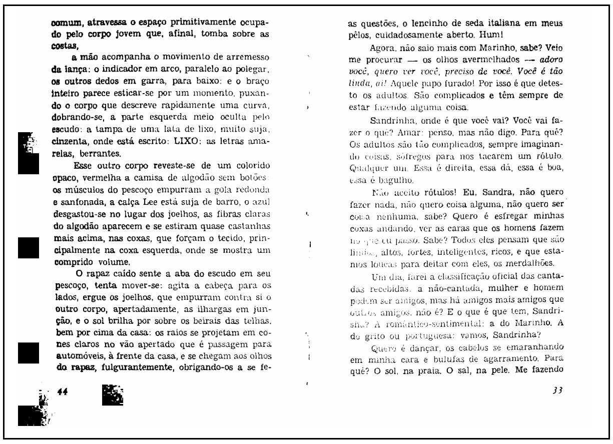
Não aceito rótulos! Eu, Sandra, não quero fazer nada, não quero coisa alguma, não quero ser coisa nenhuma, sabe? Quero é esfregar minhas coxas andando, ver as caras que os homens fazem no que eu passo. Sabe? Todos eles pensam que são lindos, altos, fortes, inteligentes, ricos, e que estamos loucas para deitar com eles, os merdalhões.

Um dia, farei a classificação oficial das cantadas recebidas: a não-cantada, mulher e homem podem ser amigos, mas há amigos mais amigos que outros amigos, não é? E o que é que tem, Sandrisha? A romântico-sentimental: a do Marinho. A do grito ou portuguesa: vamos, Sandrinha?

Quero é dançar, os cabelos se emaranhando em minha cara e bulufas de agarramento. Para quê? O sol, na praia. O sal, na pele. Me fazendo

33

Figure 12. Image binarized through the Kapur-Sahoo-Wong algorithm (18 regions).



comum, atravessa o espaço primitivamente ocupado pelo corpo jovem que, afinal, tomba sobre as costas,

a mão acompanha o movimento de arremesso da lança: o indicador em arco, paralelo ao polegar, os outros dedos em garra, para baixo: e o braço inteiro parece esticar-se por um momento, puxando o corpo que descreve rapidamente uma curva, dobrando-se, a parte esquerda meio oculta pelo escudo: a tampa de uma lata de lixo, muito suja, cinzenta, onde está escrito: LIXO: as letras amarelas, berrantes.

Esse outro corpo reveste-se de um colorido opaco, vermelha a camisa de algodão sem botões: os músculos do pescoço empurram a gola redonda e sanfonada, a calça Lee está suja de barro, o azul desgastou-se no lugar dos joelhos, as fibras claras do algodão aparecem e se estiram quase castanhas mais acima, nas coxas, que forçam o tecido, principalmente na coxa esquerda, onde se mostra um comprido volume.

O rapaz caído sente a aba do escudo em seu pescoço, tenta mover-se: agita a cabeça para os lados, ergue os joelhos, que empurram contra si o outro corpo, apertadamente, as ilhargas em junção, e o sol brilha por sobre os beirais das telhas, bem por cima da casa: os raios se projetam em cones claros no vão apertado que é passagem para automóveis, à frente da casa, e se chegam aos olhos do rapaz, fulgurantemente, obrigando-os a se fe-

44

as questões, o lencinho de seda italiana em meus pêlos, cuidadosamente aberto. Hum!

Agora, não saio mais com Marinho, sabe? Veio me procurar — os olhos avermelhados — *adoro você, quero ver você, preciso de você. Você é tão linda, ai!* Aquele papo furado! Por isso é que detesto os adultos. São complicados e têm sempre de estar fazendo alguma coisa.

Sandrinha, onde é que você vai? Você vai fazer o quê? Amar: penso, mas não digo. Para quê? Os adultos são tão complicados, sempre imaginando coisas, sôfregos para nos tacarem um rótulo. Qualquer um. Essa é direita, essa dá, essa é boa, essa é bagulho.

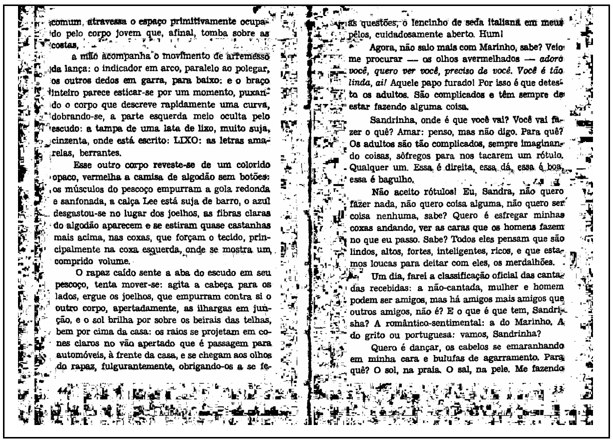
Não aceito rótulos! Eu, Sandra, não quero fazer nada, não quero coisa alguma, não quero ser coisa nenhuma, sabe? Quero é esfregar minhas coxas andando, ver as caras que os homens fazem no que eu passo. Sabe? Todos eles pensam que são lindos, altos, fortes, inteligentes, ricos, e que estamos loucas para deitar com eles, os merdalhões.

Um dia, farei a classificação oficial das cantadas recebidas: a não-cantada, mulher e homem podem ser amigos, mas há amigos mais amigos que outros amigos, não é? E o que é que tem, Sandrisha? A romântico-sentimental: a do Marinho. A do grito ou portuguesa: vamos, Sandrinha?

Quero é dançar, os cabelos se emaranhando em minha cara e bulufas de agarramento. Para quê? O sol, na praia. O sal, na pele. Me fazendo

33

Figure 13.Binarized with Sauvola (global).



comum, atravessa o espaço primitivamente ocupado pelo corpo jovem que, afinal, tomba sobre as costas,

a mão acompanha o movimento de arremesso da lança: o indicador em arco, paralelo ao polegar, os outros dedos em garra, para baixo: e o braço inteiro parece esticar-se por um momento, puxando o corpo que descreve rapidamente uma curva, dobrando-se, a parte esquerda meio oculta pelo escudo: a tampa de uma lata de lixo, muito suja, cinzenta, onde está escrito: LIXO: as letras amarelas, berrantes.

Esse outro corpo reveste-se de um colorido opaco, vermelha a camisa de algodão sem botões: os músculos do pescoço empurram a gola redonda e sanfonada, a calça Lee está suja de barro, o azul desgastou-se no lugar dos joelhos, as fibras claras do algodão aparecem e se estiram quase castanhas mais acima, nas coxas, que forçam o tecido, principalmente na coxa esquerda, onde se mostra um comprido volume.

O rapaz caído sente a aba do escudo em seu pescoço, tenta mover-se: agita a cabeça para os lados, ergue os joelhos, que empurram contra si o outro corpo, apertadamente, as ilhargas em junção, e o sol brilha por sobre os beirais das telhas, bem por cima da casa: os raios se projetam em cones claros no vão apertado que é passagem para automóveis, à frente da casa, e se chegam aos olhos do rapaz, fulgurantemente, obrigando-os a se fe-

as questões, o lencinho de seda italiana em meus pêlos, cuidadosamente aberto. Hum!

Agora, não saio mais com Marinho, sabe? Veio me procurar — os olhos avermelhados — *adoro você, quero ver você, preciso de você. Você é tão linda, ai!* Aquele papo furado! Por isso é que detesto os adultos. São complicados e têm sempre de estar fazendo alguma coisa.

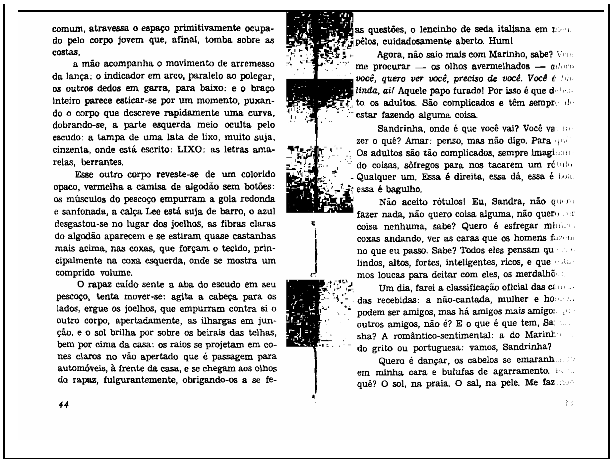
Sandrinha, onde é que você vai? Você vai fazer o quê? Amar: penso, mas não digo. Para quê? Os adultos são tão complicados, sempre imaginando coisas, sôfregos para nos tacarem um rótulo. Qualquer um. Essa é direita, essa dá, essa é boa, essa é bagulho.

Não aceito rótulos! Eu, Sandra, não quero fazer nada, não quero coisa alguma, não quero ser coisa nenhuma, sabe? Quero é esfregar minhas coxas andando, ver as caras que os homens fazem no que eu passo. Sabe? Todos eles pensam que são lindos, altos, fortes, inteligentes, ricos, e que estamos loucas para deitar com eles, os merdalhões.

Um dia, farei a classificação oficial das cantadas recebidas: a não-cantada, mulher e homem podem ser amigos, mas há amigos mais amigos que outros amigos, não é? E o que é que tem, Sandrisha? A romântico-sentimental: a do Marinho. A do grito ou portuguesa: vamos, Sandrinha?

Quero é dançar, os cabelos se emaranhando em minha cara e bulufas de agarramento. Para quê? O sol, na praia. O sal, na pele. Me fazendo

Figure 14. Binarized with Niblack (local, window_size = 50, k = -0.02)



comum, atravessa o espaço primitivamente ocupado pelo corpo jovem que, afinal, tomba sobre as costas,

a mão acompanha o movimento de arremesso da lança: o indicador em arco, paralelo ao polegar, os outros dedos em garra, para baixo: e o braço inteiro parece esticar-se por um momento, puxando o corpo que descreve rapidamente uma curva, dobrando-se, a parte esquerda meio oculta pelo escudo: a tampa de uma lata de lixo, muito suja, cinzenta, onde está escrito: LIXO: as letras amarelas, berrantes.

Esse outro corpo reveste-se de um colorido opaco, vermelha a camisa de algodão sem botões: os músculos do pescoço empurram a gola redonda e sanfonada, a calça Lee está suja de barro, o azul desgastou-se no lugar dos joelhos, as fibras claras do algodão aparecem e se estiram quase castanhas mais acima, nas coxas, que forçam o tecido, principalmente na coxa esquerda, onde se mostra um comprido volume.

O rapaz caído sente a aba do escudo em seu pescoço, tenta mover-se: agita a cabeça para os lados, ergue os joelhos, que empurram contra si o outro corpo, apertadamente, as ilhargas em junção, e o sol brilha por sobre os beirais das telhas, bem por cima da casa: os raios se projetam em cones claros no vão apertado que é passagem para automóveis, à frente da casa, e se chegam aos olhos do rapaz, fulgurantemente, obrigando-os a se fe-

44

as questões, o lencinho de seda italiana em

pêlos, cuidadosamente aberto. Hum!

Agora, não saio mais com Marinho, sabe?

me procurar — os olhos avermelhados —

você, quero ver você, preciso de você. Você

linda, ai! Aquele papo furado! Por isso é que

to os adultos. São complicados e têm sempr

estar fazendo alguma coisa.

Sandrinha, onde é que você vai? Você va

zer o quê? Amar: penso, mas não digo. Para

Os adultos são tão complicados, sempre imagi

do coisas, sôfregos para nos tacarem um ró

Qualquer um. Essa é direita, essa dá, essa é

essa é bagulho.

Não aceito rótulos! Eu, Sandra, não q

fazer nada, não quero coisa alguma, não quer

coisa nenhuma, sabe? Quero é esfregar mi

coxas andando, ver as caras que os homens f

no que eu passo. Sabe? Todos eles pensam q

lindos, altos, fortes, inteligentes, ricos, e que

mos loucas para deitar com eles, os merdalhõ

Um dia, farei a classificação oficial das c

das recebidas: a não-cantada, mulher e ho

podem ser amigos, mas há amigos mais amigo

outros amigos, não é? E o que é que tem, Sa

sha? A romântico-sentimental: a do Marinh

do grito ou portuguesa: vamos, Sandrinha?

Quero é dançar, os cabelos se emaranh

em minha cara e bulufas de agarramento.

quê? O sol, na praia. O sal, na pele. Me faz

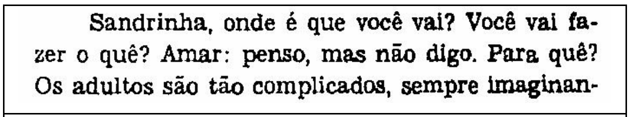
Figure 15.Binarized with MROtsu

PhotoDoc provides to the user two different ways of performing image binarization. The first one is "Automatic" and the second one opens a menu with all the algorithms above that may work either in global or image segmented mode with 2, 4, 8 and 16 regions. Local algorithm such as Niblack allows the user to define the parameters. The current implementation of "Binarization + Automatic" is set to run the algorithm by da Silva *et al.*(global) [10]. Later implementations may encompass a statistical analysis of images to choose the most suitable filter to a given image.

7. OCR

Optical Character Recognition is one of the key functionalities in any modern document processing environment [4]. Even when the recognition output is poor it may provide enough information for document indexing end keyword search. Reference [8] assesses the quality of the transcription of document images acquired with portable digital cameras with Omnipage Professional Edition version 15.0 [21]. It shows that the performance of the transcription of 50 of such documents obtained with the same models of Sony cameras used herein is close to the performance of the 100 dpi scanned counterparts.

ImageJ allows the call of executable code from within. The Tesseract [22] is an OCR Engine developed at HP Labs between 1985 and 1995. It was one of the top 3 engines in the 1995 UNLV Accuracy test. Since then, it has had little work done on it, but it is probably one of the most accurate open source OCR engines available today. The source code reads a binary, grey or color image and output text. PhotoDoc OCR whenever chosen activates the Tesseract OCR Engine. Preliminary tests such as the one made by submitting the image in the top part of Figure 15 provided as output the text in the bottom part of Figure 15.


Sandrinha, onde é que você vai? Você vai fa-
zer o quê? Amar: penso, mas não digo. Para quê?
Os adultos são tão complicados, sempre imaginan-

Sandrinha, onde E que voce vai? Voce val fa-
zer o que? Amar: penso, mas nao digo. Pars. que?
Os adultos sin tao complicados, sempre imaginan-

Figure 15 – Top: segment of textual image
Bottom: Transcribed text by Tesseract OCR

One should remark that better transcription could possibly be obtained if a Portuguese dictionary were used in conjunction with the Tesseract OCR. According to the measure of OCR performance presented in [19] the transcription above reached the figures presented on Table I.

TABLE I
ORIGINAL CHARACTER AND WORD ERRORS FOUND IN IMAGES

Character	**Replacement**	1	**Word**	**Errors**	9
	Punctuation	1		**Exclusions**	0
	G. Accents	8			
	Missing	0			
	Insertion	0			

From Table I one may see that the transcription errors were simple to be corrected as neither words nor characters are missing and there is no character insertion in the text. Besides that, word errors appeared in isolation and no word presented more than one character error in it. Most errors were due to the absence of Graphical Accents.

Conclusions and Lines for Further Works

PhotoDoc is a user friendly tool for processing document images acquired using portable digital cameras. Its current version was developed as a plug-in in ImageJ an open source portable Java library. PhotoDoc runs on users' PC and is device and manufacturer independent, working suitably from low end images acquired using cameras embedded in cell phones and palmtops to better models such as medium range, 3 and 4 Mpixels cameras, or even the state-of-the-art 6 Mpixels devices.

Several lines may be followed to provide further improvements to PhotoDoc filters. Most possibly, the most important of them is studying ways of compensating uneven illumination in documents. In the case of cameras with embedded strobe flash this may be simpler because the light source emanates from a specific point. In the case of devices that have no embedded flash illumination is provided by the environment and may come from different sources in position and power, increasing the complexity of its compensation. On the OCR front, much may be done ranging from providing dictionary help to Tesseract to performing post-processing in the textual output. This feature is already part of the newest (Jul/2007) version of Tesseract. Other open source OCR´s may also be analyzed, tested and incorporated to PhotoDoc. Preliminary testing with the open-source OCR engine OCRopus [24] showed that it was outperformed by the Tesseract OCR. For conclusive results, further testing is needed.

The PhotoDoc code is freely available at: http://www.telematica.ee.ufpe.br/sources/PhotoDoc

Acknowledgements

The work reported herein was partly sponsored by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazilian Government.

References

[1] D.Doermann, J.Liang, H. Li, "Progress in Camera-Based Document Image Analysis," ICDAR'03, V(1): 606, 2003.
[2] J. Liang, D. Doermann and H. Li. Camera-Based Analysis of Text and Documents: A Survey. International Journal on Document Analysis and Recognition, 2005.
[3] K.C.Fan, Y.K.Wang, T.R.Lay, Marginal noise removal of document images, Patt.Recognition. 35, 2593-2611, 2002.
[4] Lu S and C L Tan, Camera document restoration for OCR, CBDAR 2005/ICDAR 2005, Seoul, Korea.
[5] L.G.Shapiro and G.C.Stockman, Computer Vision, March 2000. http://www.cse.msu.edu/~stockman/Book/book.html.
[6] L. Jagannathan and C. V. Jawahar, "Perspective correction methods for camera based document analysis," pp. 148–154, CBDAR 2005/ICDAR 2005, Seoul, Korea. 2005.
[7] R. Gomes e Silva and R. D.Lins. Background Removal of Document Images Acquired Using Portable Digital Cameras. LNCS 3656, p.278-285, 2005.
[8] R.D.Lins, A.R.Gomes e Silva and G.Pereira e Silva, Assessing and Improving the Quality of Document Images Acquired with Portable Digital Cameras, ICDAR´07, Curitiba, Brasil, 2007.
[9] H.S.Baird, Document image defect models and their uses, ICDAR'93, Japan, IEEE Comp. Soc., pp. 62-67, 1993.
[10] J. M. M. da Silva *et al.* Binarizing and Filtering Historical Documents with Back-to-Front Interference, ACM-SAC 2006, Nancy, April 2006.
[11] T. Pun, Entropic Thresholding, A New Approach, C. Graphics and Image Processing, 16(3), 1981.

[12] J. N. Kapur, P. K. Sahoo and A. K. C. Wong. A New Method for Gray-Level Picture Thresholding using the Entropy of the Histogram, Computer Vision, Graphics and Image Processing, 29(3), 1985.

[13] L. U. Wu, M. A. Songde, and L. U. Hanqing, An effective entropic thresholding for ultrasonic imaging, ICPR'98: Intl. Conf. Patt. Recog., pp. 1522–1524 (1998).

[14] M.R .Gupta, N.P. Jacobson and E.K. Garcia. OCR binarization and image pre-processing for searching historical documents. Pattern Recognition 40 (2007): 389-397 (2007).

[15] J. C. Yen,*et al.* A new criterion for automatic multilevel thresholding. IEEE T. Image Process. IP-4, 370–378 (1995).

[16] G. Johannsen and J. Bille. A threshold selection method using information measures. ICPR'82: 140–143 (1982).

[17] W.Niblack, "An Introduction to Image Processing" pp.115-116, Prentice-Hall,Englewood Cliffs,NJ(1986).

[18] J. Sauvola, M. Pietikainen, Adaptive document image binarization, Pattern Recognition 33 (2) (2000) 225–236.

[19] R.D.Lins and N.F.Alves. A New Technique for Assessing the Performance of OCRs. IADIS – Int. Conf. on Comp. Applications, IADIS Press, v. 1, p. 51-56, 2005.

[20] ImageJ http://rsb.info.nih.gov/ij/

[21] Nuance Corp. http://www.nuance.com/omnipage/professional

[22] Tesseract http://code.google.com/p/tesseract-ocr/

[23] JAI (Java Advanced Imaging). https://jai.dev.java.net.

[24] OCRopus .http://www.ocropus.org.

Effective Feature Extraction Using Low-Resolution Images

Hirotaka OHTA and Kazuhiko YAMAMOTO
Department of Information Science, Faculty of Engineering, Gifu University
ohta@yam.info.gifu-u.ac.jp

Abstract

When character recognition is made from low-resolution characters of motion image, it is the general idea to restructure high-resolution image first by using sequence of the low-resolution images and then extract features from constructed high-resolution image. In this paper, we propose a new method in which the direct extraction of features from the low-resolution images is made first, and then reconstructing high accuracy feature from sequence of feature. We show the advantage of our proposed method for ordinary method in comparative recognition experiment both for ideal database images.

1. Introduction

In recent years, the digital video camera has penetrated to the general people. Images taken by digital video cameras often suffer from a lack of sufficient resolution. Therefore, it is difficult to extract features effectively for the character recognition from these images. There are two competing approaches to solving this problem.

In the first approach, the feature image is extracted from the high-resolution image that is reconstructed from the low-resolution images. The method of the reconstructed high-resolution image has been proposed by many researchers [1]~[4]. The correspondence information (rotation, shift, magnification etc.) between each image is necessary to reconstruct the high-resolution image. However, the method of using the correspondence information to recognize by using reconstructed image is not researched so much.

In the second approach, the feature images are extracted directly from the low-resolution images [5],[6].

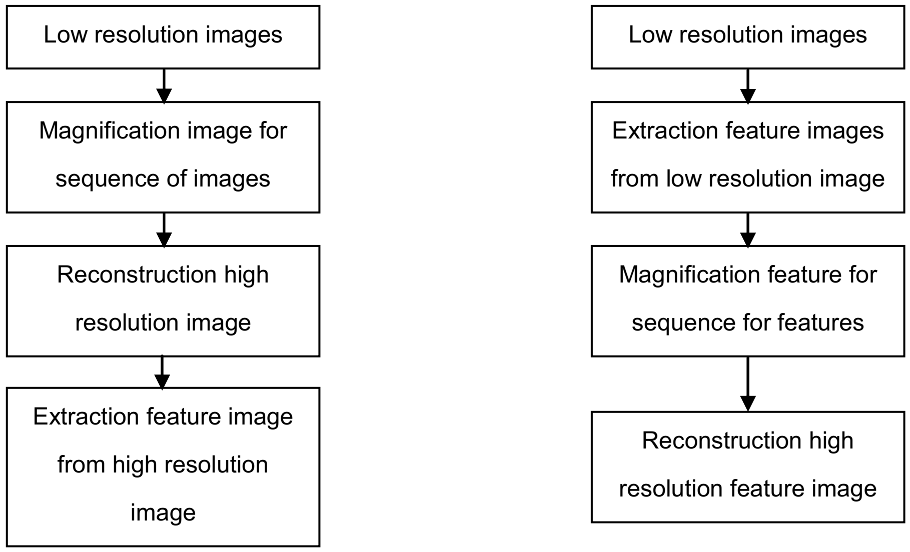
In this paper, we propose a new processing method: extracting edge features first and reconstructing the high-resolution feature images. We test this method against the most ordinary method; reconstructing the high-resolution image by the use of sub-pixels first and extracting feature images from high-resolution image. We show that our proposed method gives higher accuracy than the alternative recognition method for image sequence of low-resolution.

2. Linear problem between two methods

In this section, we explain processes of our proposed method and the ordinary method. Processes of two methods have difference of the stage for extracting features. Figure 1 is shown our proposed method process and the ordinary method process.

Input images of the actual experiment are used two-dimensional images. It is difficult to explain the different phenomenon of each method by the two-dimensional image. The phenomenon on each method appears with the one-dimensional model image. In this paper, these methods are explained for easily by the one-dimensional model images. Figure 2 is an example one-dimensional model of input images consisted of 5×1 resolution. In each image A and C, the center of gravity shifted by 0.5 pixels from the center image B.

For example, image A shifted to the left by 0.5 pixels from image B. By superposing and summing up these images of 0.5 pixels shifted, it can reconstruct a double resolution image. To extract features from these images that have sub-pixel shift, the general idea reconstructs the high-resolution image from these low-resolution images and extracts the four directional feature fields.


Low resolution images
Magnification image for sequence of images
Reconstruction high resolution image
Extraction feature image from high resolution image
Low resolution images
Extraction feature images from low resolution image
Magnification feature for sequence for features
Reconstruction high resolution feature image

(a) The ordinary method on feature extraction from low resolution images

(b) Propose method on feature extraction from low resolution images

Figure 1. The process of feature extraction from sequence for low resolution images

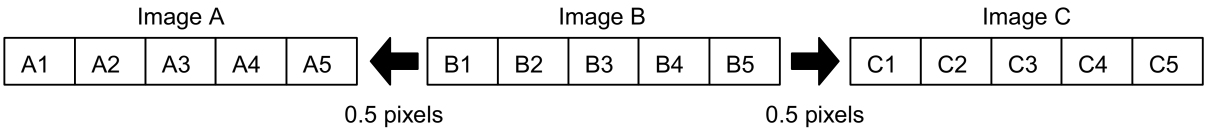


Figure 2. One dimensional model of input images

In the 5×1 image of figure 2, "A1, A2, A3 …", "B1, B2, B3 …", and "C1, C2, C3 …" mean each pixel value. Each image is magnified double size by using nearest neighbourhood method as shown in figure 3. By superposing and summing up each image of figure 3, the high-resolution image of 12×1 is reconstructed as shown in figure 4. An equation inside of each frame means the pixel value. For example when x=5, y=0, value of (x, y) is A2+B3+C3, it is calculated from pixel values of figure 2.

By using this reconstructed image, the existing method extracts the feature through the filter as shown in figure 5. In figure 4, a edge feature in a frame indicated by position (x, y)=(5,0) is expressed by the absolute values of subtraction of pixel values "(x, y)=(4,0) and (x, y)=(6,0)". It is shown by eq.(1).

$$|\ (\ A2+B2+C3\ ) - (\ A3+B3+C4\ )\ | \qquad (1)$$

On the other hand, in our proposed method, first, the features are extracted from figure 2 images under the low-resolution through the filter as show in figure 6. After each extracted feature images is magnified double size by using nearest neighbourhood as shown in figure 7. The high resolution feature image is created by superposed these edge features. Here, the summed up equation in the frame indicated by dashed line is expressed by eq.(2).

$$|\ A2\text{-}\mathrm{A}3\ | + |\ B2\text{-}B3\ | + |\ C3\text{-}C4\ | \qquad (2)$$

When compare between eq.(1) and eq.(2), there is difference about the stage of the calculation of absolute value. The edge feature is affected by its difference. The edge feature to extract from the reconstructed high resolution image is smoothed to add the pixel value of each low resolution image in equal condition. Therefore, the influence of the edge feature to exist on each low resolution image is reduced in some cases. In the ordinary method, the existence of the edge feature is caused degeneration. The proposed method is constructed only summed up the edge feature that is extracted each low resolution image. The existence of the edge feature isn't caused degeneration.

The influence by these differences is shown to the recognition object of the complex shape, because the degeneration is shown around the pixel of the edge feature. Therefore, our proposed method can create more effective edge feature image which is effective for the recognition than the ordinary method.

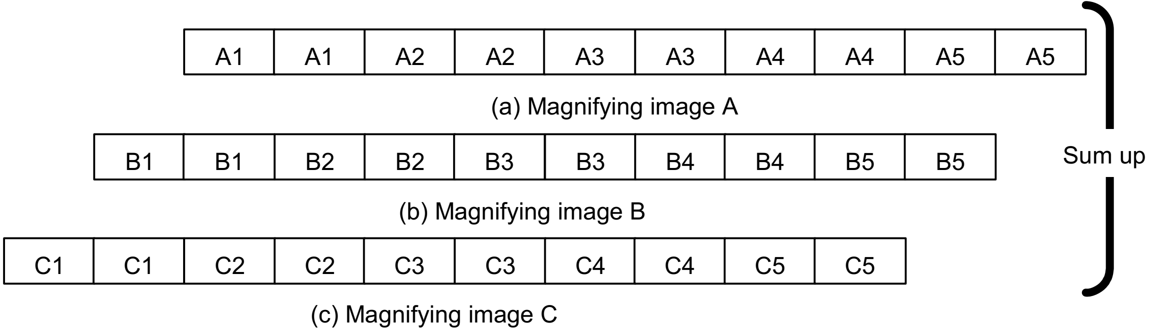
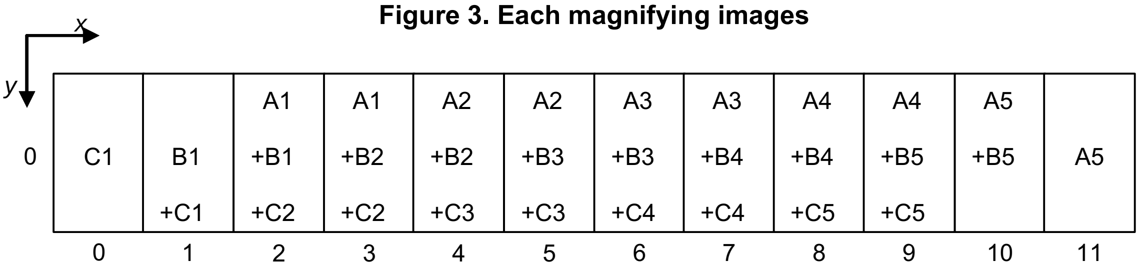

A1 A1 A2 A2 A3 A3 A4 A4 A5 A5
(a) Magnifying image A
B1 B1 B2 B2 B3 B3 B4 B4 B5 B5
(b) Magnifying image B
C1 C1 C2 C2 C3 C3 C4 C4 C5 C5
(c) Magnifying image C
Sum up

Figure 3. Each magnifying images



	0	1	2	3	4	5	6	7	8	9	10	11
0	C1	B1 +C1	A1 +B1 +C2	A1 +B2 +C2	A2 +B2 +C3	A2 +B3 +C3	A3 +B3 +C4	A3 +B4 +C4	A4 +B4 +C5	A4 +B5 +C5	A5 +B5	A5

Figure 4. Reconstructed high resolution image

1	0	-1

Figure 5. The filter of feature extraction for high resolution image

1	-1

Figure 6. The filter of feature extraction for low resolution image

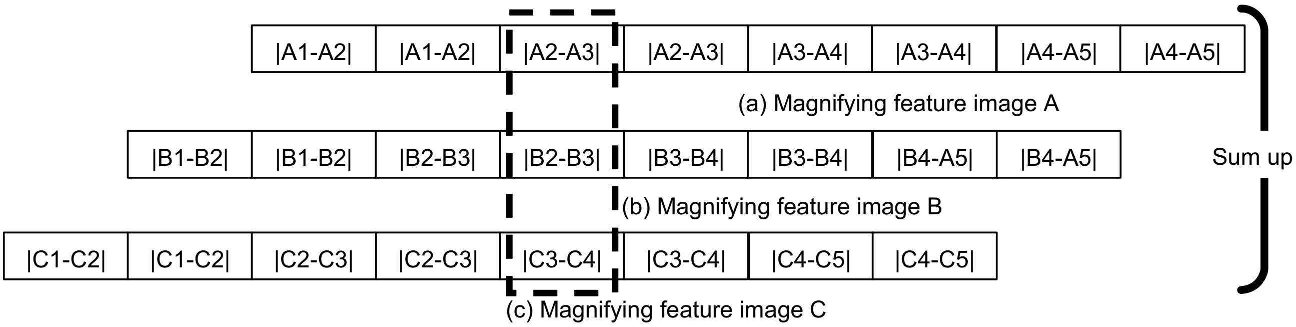

|A1-A2| |A1-A2| |A2-A3| |A2-A3| |A3-A4| |A3-A4| |A4-A5| |A4-A5|
(a) Magnifying feature image A
|B1-B2| |B1-B2| |B2-B3| |B2-B3| |B3-B4| |B3-B4| |B4-A5| |B4-A5|
(b) Magnifying feature image B
|C1-C2| |C1-C2| |C2-C3| |C2-C3| |C3-C4| |C3-C4| |C4-C5| |C4-C5|
(c) Magnifying feature image C
Sum up

Figure 7. Each magnifying feature images

3. Features extraction method

3-1. Four directional features field

The feature in this paper is used four direction features field [7]. This feature is characterized by four direction edges (horizontal, right-up, vertical and right-down) from the input image. In this paper, the extracted feature from the low resolution image is used 2x2 size filter as shown in figure 8. And, the extracted feature from the high resolution image is used 3x3 size filter as shown in figure 9.

1	0
-1	0

Horizontal

0	1
-1	0

Upper-right

1	-1
0	0

Vertical

1	0
0	-1

Lower-right

Figure 8. The feature extraction filters through the low resolution image

1	0	0
0	0	0
-1	0	0

Horizontal

0	0	1
0	0	0
-1	0	0

Upper-right

1	0	-1
0	0	0
0	0	0

Vertical

1	0	0
0	0	0
0	0	-1

Lower-right

Figure 9. The feature extraction filters through the high resolution image

3-2. Feature extraction method of our proposed method

Our proposed method uses a different order of operations compared to the ordinary method. Our method extracts features from low-resolution input images first, and creates the high-resolution feature. To use this method it is necessary that an input image sequence has sub-pixel shift of the center of gravity. For the ordinary experiment, we made a basis image by scanning as shown in figure 10. And then, we made eight ideal images from the basis image by shifting 1 pixel and reducing to half-resolution theoretically. Thus we made images that shifted 0.5 pixels in eight directions, as shown in figure 11. Figure 12 shows the four directional features field by using our proposed method.

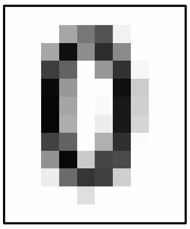


Figure 10. Scanning high resolution

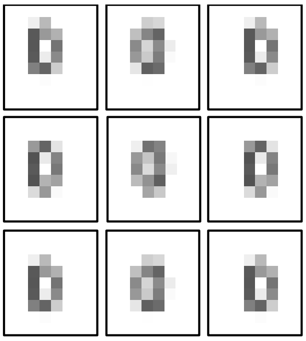


Figure 11. Ideal low resolution images to shift by sub-pixels on eight directions

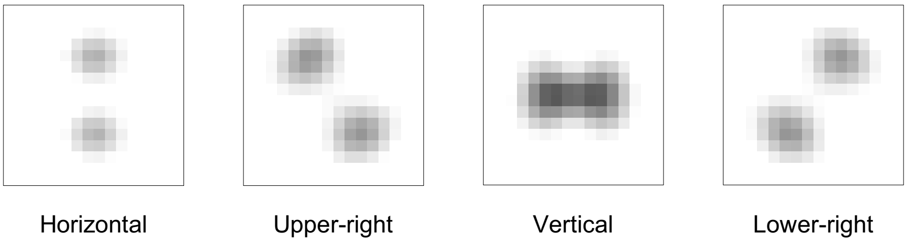


Figure 12. Example of the feature image by using our propose method

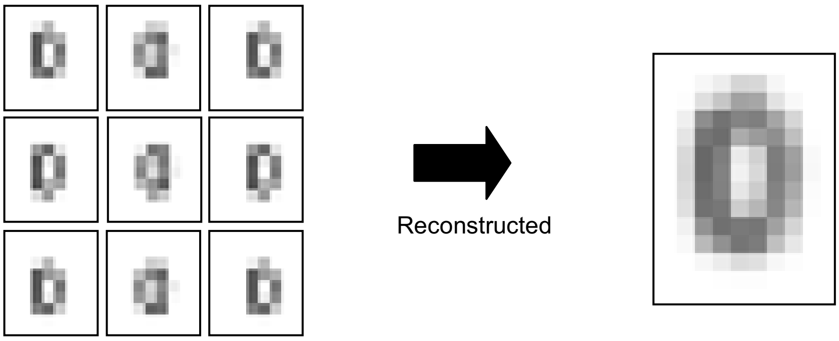


Figure 13. Reconstructed high resolution image from ideal low resolution images

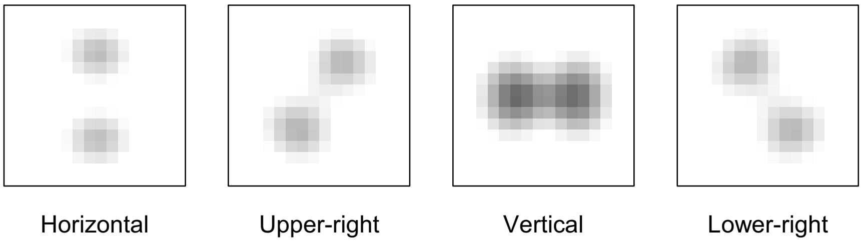


Figure 14. Example of the feature images by using ordinary method

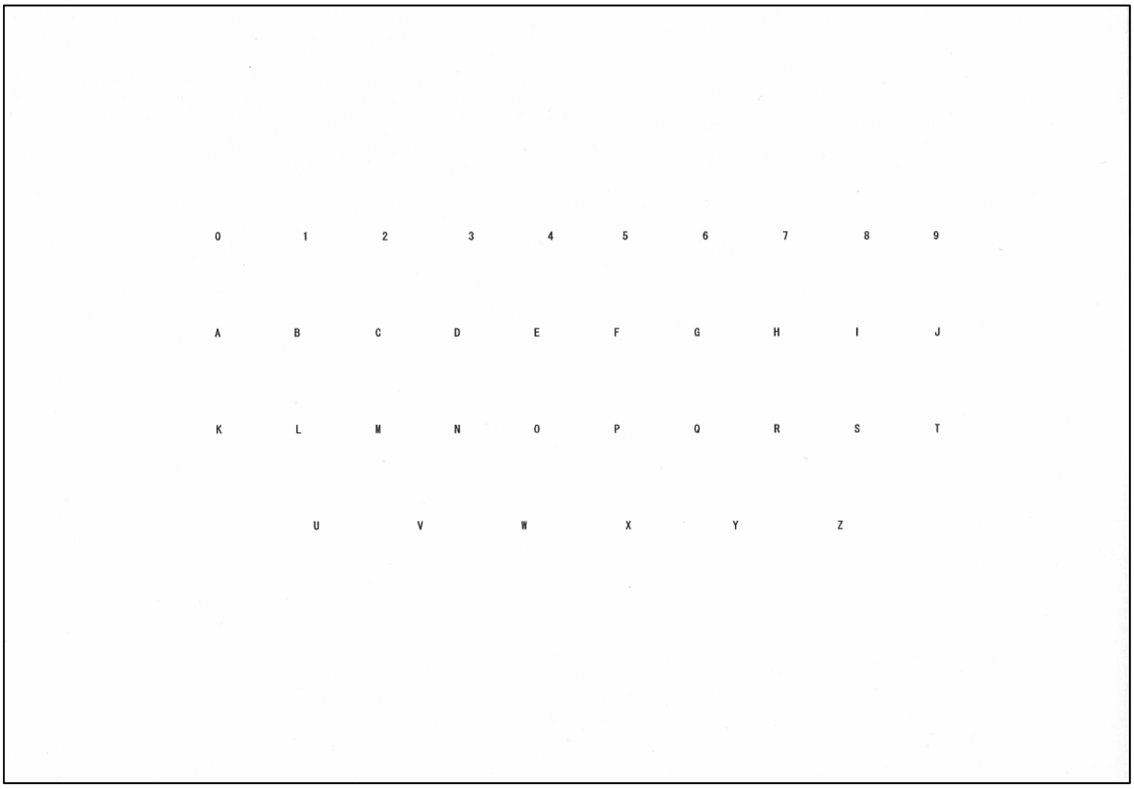


Figure 15. Example of scanning image (8pt, 100dpi)

3-3. Feature extraction method of the ordinary method

T In the ordinary method, we reconstructed the high-resolution image as shown in figure 13 with superimposing the sub-pixel shift of the center of gravity of each images from ideal low-resolution images as shown figure 11. Then, four directional features field as shown in figure 14 is extracted from the high-resolution image.

When compare between figure 12 and figure 14, there is difference, but difficult to find out exact advantage of our method. Then we need comparative recognition experiment.

4. Comparative recognition experiment on database images

4-1. Database images and creating dictionary

The database consisted of 36 characters (0 through 9 and A through Z in the MS Gothic typeface). These were printed on A4 plain paper with an inkjet printer. We took 200 sets of basis images at three font sizes (8 pt., 10 pt., and 12 pt.) with a scanner in 100dpi. Figure 15 show the example of the scanning image. After reducing to low-resolution and making input images, 100 sets were used for making the dictionary, and other 100 sets were used as unknown input images. Then, we made dictionaries by each method (horizon, upper right, vertical and lower right). These are taken average each direction feature of 100 sets. Figure 16 shows examples of database images.

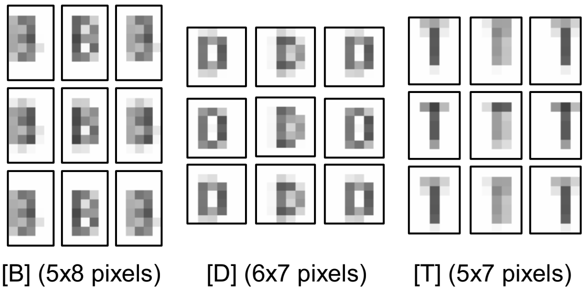

[B] (5x8 pixels)   [D] (6x7 pixels)   [T] (5x7 pixels)

Figure 16. Example of database images (8pt, 100dpi)

4-2. Process of experiment

Input images were subjected to some pre-processing, removal of back ground, noise reduction and segmentation. After extracting features by several methods, dictionaries were created by normalization using the Gaussian filter. The four directional features field are extracted by same process, and characters were extracted.

4-3. Experimental result

We experimented comparing between our proposed and the ordinary method. Results of recognition rate are shown in table 1 and figure 17.

Table 1. Experiment result

	Font size		
	12pt	10pt	8pt
Ordinary method (%)	96.08	92.00	87.83
Proposed method (%)	96.94	95.47	90.75

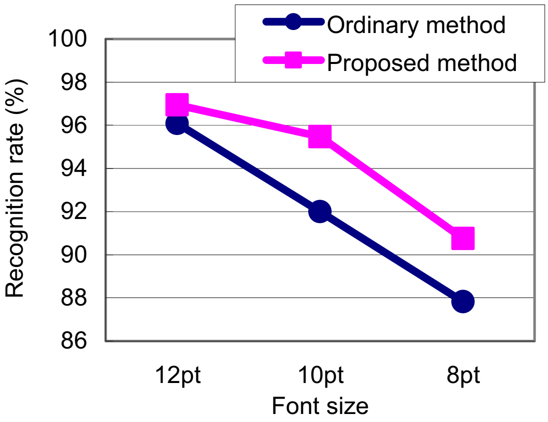


Figure 17. Experiment result on data-base

As show in table 1, our proposed method gives higher accuracy than the ordinary method. That reason why the ordinary method was influenced by the degeneration of the directional vector component by taking the absolute value. In the ordinary method the information in the direction of input images is lost when the high-resolution image is reconstructed. Directional vector component is not extracted from the high-resolution image. On the other hand, the feature of our proposed method is not influenced by taking the absolute value, because edge feature is made extracting from low-resolution and superimposing these images. Therefore the proposed method gives higher accuracy and the edge directional feature extracted is better than that in the ordinary method.

5. Conclusion

We proposed new method of the feature extraction that is effective with low resolution image. It was shown that there is nonlinearity between two methods which inverse order of procedure of derivation and integration. The proposed method is more effective than ordinary method for low resolution character recognition through experiment for the database images. Our proposed method is less influence of degeneration by using the original value until extracting the feature. We showed the performance of our proposed method for database images. Future works is experiment that increases the number of database and category.

Acknowledgment

This research was supported in part by the Grant-in-Aid for the WABOT-HOUSE Project by Gifu Prefecture.

References

[1] S.C.Park, M.K.Park, and M.G.Kang, "Super-Resolution Image Reconstruction: A Technical Overview", IEEE Signal Processing Magazine, Vol.20, No.3, pp.21-36, 2003.

[2] S.Baker and T.Kanade, "Hallucinating Faces", Proc. of the Fourth International Conference on Automatic Face and Gesture Recognition 2000, pp.83-88, 2000.

[3] B. C. Tom and A. K. Katsaggelos: “Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images”, Proc. of International Conference on Image Processing, vol.2, pp.2539-2542, 1995.

[4] R. Schultz and R. Stevenson: “Extraction of High Resolution Frames from Video Sequences”, IEEE Trans. Image Process, Vol.5, No.6, pp.996-1011, 1996.

[5] H.Ishida, S.Yanadume, T.Takahashi, I.Ide, Y.Mekada, and H.Murase, "Recognition of low-resolution characters by a generative learning method", Proc. of the First International Workshop on Camera-Based Document Analysis and Recognition, Vol.O2-2, pp.45-51, 2005.

[6] J. Kosai, K. Kato, and K. Yamamoto “Recognition of Low Resolution Character by a Moving Camera”, Proc. of International Conference on Quality Control by Artificial Vision, pp.203-208, 1999.

[7] K.Yamamoto, “Present State of Recog-nition Method on Consideration of Neighbor Points and Its Ability in Common Database”, IEICETrans. Inf. & Syst. Vol.E79-D, No.5, pp.417 422, 1996.

Text Detection in Natural Scene Images using Spatial Histograms

Shehzad Muhammad Hanif, Lionel Prevost
Université Pierre et Marie Curie Paris 06, ISIR FRE 2507
BC 252, 4 Place Jussieu 75252 Paris CEDEX 05, France
shehzad.muhammad@lisif.jussieu.fr, lionel.prevost@upmc.fr

Abstract

In this paper, we present a texture-based text detection scheme for detecting text in natural scene images. This is a preliminary work towards a complete system of text detection, localization and recognition in order to help visually impaired persons. We have employed spatial histograms computed from gray-level co-occurrence matrices for texture coding and three classifiers have been evaluated. Moreover, one feature selection mechanism is employed to select those histogram components that exhibit high discrimination power. The texture coding scheme is simple and can readily differentiate between text and non-text. The proposed scheme is evaluated on 50 images taken from ICDAR 2003 robust reading and text locating database. The results are encouraging with a text detection rate of 66% and a false alarms rate of 22% .

1. Introduction

This work is a part of the project called "Intelligent Glasses" [1] (Figure 1). The aim of the project is to help blind and visually impaired persons to know their environment in a better way. The Intelligent Glasses are a man-machine interface which translates visual data (such as 3D global information) onto its tactile representation. It has three parts, a bank of stereovision, a processing unit for visual perception and a handheld tactile of braille surface type. The visual data are acquired and processed by the vision system, while its tactile representation is displayed on a touch stimulating surface. In its original form, this system is able to provide information about different types of obstacles and their position with respect to user. Also, it can represent different geometrical shapes (square, rectangle, circle, arcs, curves….) on its tactile interface as well as braille symbols.

The need of textual information for blind and visually impaired persons is obvious. While taking into account this need, we have added an extra module to visual perception step of the above said system that will detect, localize and recognize the text in captured images and all this textual information will be displayed on the tactile surface.

Text detection, localization and recognition in images are regarded as basic operations in processing the captured images and are a necessary part of any application of camera based document analysis. In these recent years, they have gained a lot of attention. In general, the domain is divided into two parts based on the type of text appearing in images – one deal with super-imposed text appearing in images and videos called graphic text and other deal with the text appearing in the captured scene called scene text.

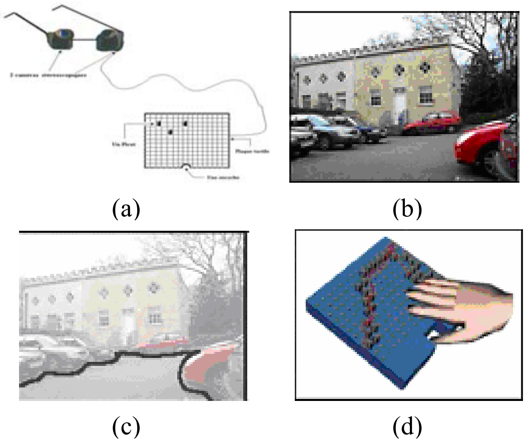

(a) (b)
(c) (d)

Figure 1. Intelligent Glasses: (a) concept (b) scene (c) environment perception (d) representation

This paper addresses the problem of scene text detection in gray-level images and presents preliminary works in this domain. Our algorithm is simple and generates potential/candidate text regions that can later be verified by a validation/verification scheme. We follow a general framework of candidate regions generation and validation as employed by various researchers [3][5][6]. We have not proposed any localization or recognition algorithm. Our detection method is based on spatial histograms computed from gray-level co-occurrence matrix (GLCM) [2]. These spatial histograms capture texture information present in the image. Based on the fact that text is a distinct texture, we can distinguish between text and non-text regions. Our classification scheme classifies image pixels as text or non-text. Furthermore, connected components can be extracted and each of them can be verified by the validation/verification scheme. In this paper, we have employed three different classifiers.

The rest of the paper is organized as follows. Section 2 contains overview of existing texture based text detection methods. Section 3 describes the construction of GLCM and computation of spatial histograms along with different classifiers used for classification. In section 4, natural scene image database and text detection results are described. Section 5 concludes this paper. Various prospects are also discussed in the section.

2. Overview of existing methods

Existing methods for text detection, localization and extraction can broadly be classified as gradient features based, color segmentation based and texture features based [8]. Here, we will concentrate on texture methods. Text is viewed as a unique texture that exhibits a certain regularity that is distinguishable from background. Humans can identify text of foreign languages even when they do not understand them largely due to its distinct texture. Various researchers have exploited this fact to detect text in images. The texture methods are largely used for text detection. Texture features can be extracted directly from the pixel's spatial relationships or from frequency data.

Wu et al.[3] proposed a texture segmentation method based on linear filtering using nine second derivatives of Gaussians filters at different scales to generate candidate text regions. The output of each filter is passed through a sigmoid transfer function. For each pixel location, local energy serves as feature vector for the pixel. The set of feature vectors is clustered using K-means algorithm. A verification algorithm is proposed by the authors to filter text-like regions.

Jung et al. [4] employed a multi-layer perceptron (MLP) classifier to discriminate between text and non-text pixels. A sliding window scans the whole image and serves as the input to neural network. Each center pixel of the window is classified as text or non-text pixel. The output image serves as a probability map where high probability areas are regarded as candidate text regions.

In [5], Crandell et al. have used a sophisticated approach to select those DCT coefficients that can distinguish text and non-text regions. They proposed text detection, localization, binarization and text tracking schemes to detect caption text from color video sequences of television channels. The scheme is claimed to work also on high contrast scene text. Text detection is based on text energy defined as sum of absolute of DCT coefficients. A subset of 19 DCT coefficients is chosen empirically by selecting coefficients with high discrimination power.

Gllavata et al. [6] used wavelet transform to perform texture analysis for text detection. A filter bank is used to extract low and high frequency sub-bands. These high frequency sub-bands are used in classification step to detect text. They have used k-means algorithm to cluster text and non-text regions.

An enhanced version of previous method is applied to color images by Saoi et al. [7] for text detection in natural scene images. In this technique, a color image is decomposed into R, G and B channels. Next wavelet transform is applied to all channels separately. High frequency sub-bands are considered for feature vector generation and k-means algorithm is employed for clustering. Contrary to previous method, this clustering is applied in a combined space.

As said earlier, texture is believed to be a rich source of visual information and it is easily perceived by humans. Thus texture methods are strong candidates to be adopted for text detection task. However, these methods are often computationally expensive and are greatly dependant on contrast of the text in an image, but lead to good results.

3. Proposed method

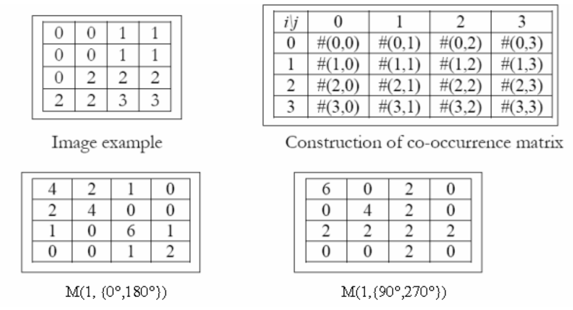
3.1. Texture coding scheme

We have proposed a simple texture coding method to detect scene text in gray-level natural scene images. We have used spatial histograms computed from gray-level co-occurrence matrix (GLCM) for texture coding. Gray level co-occurrence matrix $M_{(x,\, y)}(d,\, \theta)$ or second order histogram (which consider the relationship between groups of two pixels in the original image) was initially defined by Haralick [2]. Since then, GLCM has been widely used in remotesensing and analyzing satellite images. In most of the cases, this method is used in texture segmentation.

By simple definition, GLCM is a tabulation of how often different combinations of pixel values (gray levels) occur in an image. When divided by the total number of neighboring pixels $R_{(x,\ y)}(d,\ \theta)$ in the image, this matrix becomes the estimate of the joint probability $p_{(d,\ \theta,\ x,\ y)}(i,j)$ or $p(i,j)$ of two pixels, a distance d apart along a given direction θ having particular (co-occurring) gray values i and j. Moreover, x and y represent the spatial position of matrix. The dimension of GLCM is GxG where G is the number of gray-levels used to construct the matrix.

Generally, GLCM is computed over a small square window of size N centered at a certain pixel (x, y) and then window is moved by one pixel in the same manner like convolution kernel. Fine texture description requires small values of d and/or small window size, whereas coarse texture requires large values of d and/or large window size. An average over all orientations is taken so that these matrices are rotation invariant.

Figure 2 shows an example of construction of gray-level co-occurrence matrix for d = 1 and θ = {0°, 180°} and {90°, 270°}. The matrix $M_{(0,\ 0)}$ (1, 180°) is just the transpose of $M_{(0,\ 0)}$ (1, 0°). So to cover all orientations (8 in this case), we need only to compute first four orientations.



0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Image example

i|*j*	0	1	2	3
0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

Construction of co-occurrence matrix

4	2	1	0
2	4	0	0
1	0	6	1
0	0	1	2

M(1, {0°,180°})

6	0	2	0
0	4	2	0
2	2	2	2
0	0	2	0

M(1,{90°,270°})

Figure 2. Construction of co-occurrence matrix[2]

3.2. Spatial Histograms

It is known that feature based algorithms are generally more stable than raw data (gray levels) based algorithms so a number of features can be calculated using the co-occurrence matrix (containing G^2 elements) for texture discrimination. Haralick defined 14 such features. Among these 14 features, contrast, entropy, homogeneity, energy are commonly used for image classification purpose.

However, in present work, we do not consider these features. As stated earlier, the GLCM represent a joint distribution $p(i,j)$ of pixels, so we can also take into account the marginal probabilities $p(i)$ or $p(j)$ of this joint distribution. Moreover, the GLCM is computed in various directions and distances and it also cover spatial relationship between the pixels, so the marginal distributions must contain information about texture, shape and spatial relationship between the pixels and represent useful information about the nature of the pixels in the image. As text and non text regions differ in their texture, shape and spatial relationships, so these probabilities are useful for the classification task. Another usefulness of marginal probabilities is their compactness as GLCM contains G² entries and most of them are zeros. On the other hand, marginal probability contains only G elements. Due to symmetric nature of GLCM, both probabilities $p(i)$ and $p(j)$ are equal, so we take only one of them into account. From now onward, these marginal probabilities will be called spatial histograms.

3.3. Classification

Three types of classifiers have been employed. Two of them are discriminative and third one is generative. The objective is to find a classifier that gives low false alarms and high text detection rate.

3.3.1. Maximum a posteriori (MAP) classifier. A likelihood estimate is computed based on the assumption that the posterior probability of the data (spatial histograms) is a uni-modal multivariate gaussian. Gaussian parameters i.e. mean and covariance matrix for each class are estimated on a training database by using maximum likelihood estimator. Mathematically, discriminant function based on log likelihood can be written as:

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{G}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}_i|) + \log P(C_i)$$

for i = 1(text), 2(non-text)

where $\boldsymbol{\mu}_i$ = mean of ith class, $\boldsymbol{\Sigma}_i$ = covariance matrix of ith class, $P(C_i)$ is the prior probability of ith class (equally distributed on both classes) and G is the dimension of spatial histogram (**x**)

In MAP classifier, for a test example, we simply find the class that maximizes the posterior probability.

$$\left(\operatorname*{argmax}_i \ g_i(\mathbf{x})\right)$$

3.3.2. Neural classifier. Next, we employed a multi-layer perceptron (MLP) classifier. Spatial histograms from training database are fed to neural classifier as input and the desired outputs are example labels: 1 for text, -1 for non-text. Cross-validation is used as a stopping criterion. The number of neurons in hidden cells is optimized during experimentation.

3.3.3. Text class generative model. Finally, we employed a generative model. Text class is modeled by the mean spatial histogram (MSH) estimated on the training database. A spatial histogram corresponding to a arbitrary pixel is compared to MSH through a similarity measure or distance. If that arbitrary pixel is a text pixel, then similarity measure (distance) gives a value close to 1(0), if not, the similarity (distance) value will be closer to 0(1). A simple threshold on the similarity measure will determine example's class. However, the selection of threshold is not trivial.

Furthermore, we have observed that spatial histograms do contain some zero elements so we can employ a dimensionality reduction scheme. For dimensionality reduction, we employ principal component analysis. We can see from principal components' cumulative energy curve (Figure 3) that 13 components are required to preserve 90% of energy. However, we don't want to reconstruct original data from these components. The goal is to find those components that can help in classification i.e. have high discrimination power. So retaining 13 components having 90% cumulative energy might not be the correct choice.

Hence, we have to adapt a procedure that selects an optimal threshold and optimal number of principal components in order to maximize the text detection rate and minimize false alarm rate. For this task, we employ a feature selection mechanism; more precisely it is a wrapper method for feature selection. During training, number of principal components and threshold value are found by exhaustive searching in which the similarity measure acts as part of the classifier. The percentage of false alarms is kept fixed and text detection rate is maximized on the training database by varying the number of principal components and threshold value.

4. Experimental results

4.1. Database

We have used ICDAR 2003 robust reading and text locating database [10] in our experimentation. The trial database is divided into two parts: TrialTrain and TrialTest. However, in our experimentation, we have used a total of 100 images taken from TrialTrain part. These images contain text with various font sizes, word lengths, orientations and colors. The size of images varies from 640x480 to 1024x768. There are 433 text segments in the images and font size varies from 10 pixels to 300 pixels. Out of these 100, 50 images are used for training and other 50 for test. For training different classifiers, 100,000 text examples and 100,000 non-text examples are taken randomly from 50 images. As a preprocessing step, images are converted to gray scale. No other preprocessing is employed.

4.2. Computation of gray-level co-occurrence matrices and spatial histograms

We compute GLCMs over a small square window of size N centered at a certain pixel (x, y) and then window is moved by one pixel in convolution kernel manner. GLCMs are computed in 8 directions (E, NE, N, NW, W, SW, S, SE) or ($d = 1$, $\theta = 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°$) and an average is taken so that these features are rotation invariant. In actual implementation only four orientation matrices are needed to be calculated and the other four orientations can be computed using transpose of these matrices. Moreover, five different square windows with size N = 5x5, 7x7, 9x9, 11x11, 13x13, 17x17 are used.

Due to intensive nature of computations, reduction of number of intensity levels (quantizing the image to a few levels of intensity) helps increase the speed of computation with negligible loss of textural information. The gray-levels are uniformly quantized to G levels between minimum and maximum gray levels of a window. We choose 32 gray-levels in our implementation. Once GLCMs are computed, spatial histograms can readily be computed by summing up rows or columns.

4.3. Text detector evaluation method

To evaluate the performance of a certain text detector, we adopt a pixel based evaluation mechanism. The target images (binary in nature) in ICDAR 2003 database contains one or more black (pixel values = 0) rectangular areas representing text regions. The pixel value for non-text pixels is 1. The database is designed for text localization. However, in our scheme, due to absence of localization step which generates rectangles around text strings, we have to evaluate performance of text detector with the given target images where text regions are represented by rectangular regions and figure-ground data is missing.

The text detector generates either 0 (for text) or 1 (for non text) for each pixel of the input image. In pixel based evaluation mechanism, the output of text detector is compared with the target and a confusion matrix is created. For evaluation, two quantities, text detection rate and false alarm rate are computed.

4.4. Text detector results

In this section, we explain the training strategy and/or parameter tuning mechanism of each text detector as well as the results. Connected components can be extracted from the output binary image of the text detector in a post-processing step. Each connected component can be verified by a validation/verification scheme.

4.4.1. Maximum a posteriori (MAP) classifier. During parameter estimation of gaussian distribution, it has been observed that covariance matrices are ill-conditioned so only variances are considered (i.e. covariance matrices in the discriminant function are diagonal). However, variances are not equal. The text detector based on MAP gives a text detection rate of 72.5% and false alarm rate is 37%. The best window size is 17x17. The problem with this text detector is the high false alarms rate.

4.4.2. Neural classifier. The best neural classifier after experimentation has 32 inputs, 5 hidden cells and 2 outputs. The network was trained for 10000 epochs with cross-validation stopping. One-fourth of the training database is used in cross-validation while the rest is used for training. This text detector gives 66% text detection rate and 22% false alarm rate. The best window size is 17x17. In terms of false alarm rate, we can say that this text detector is better but text detection rate is dropped by 6%.

4.4.3. Text class generative model. As stated in §3.3.3, PCA is used for dimensionality reduction of spatial histograms. Figure 3 shows the principal components' cumulative energy curve obtained on training database for N = 17x17. It is clear that to retain 90% energy, 13 principal components are required. However as argued earlier, this may not be the correct choice for classification task.

Next we choose one similarity measure (distance) that will act as a part of the generative model. Four different similarity measures (distance), commonly cited in literature, are used to measure similarity between two spatial histograms - say $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_G\}$ - a test example and $\mathbf{MSH} = \{\mathbf{m}_1,\mathbf{m}_2,\ldots, \mathbf{m}_G\}$. These measures are:

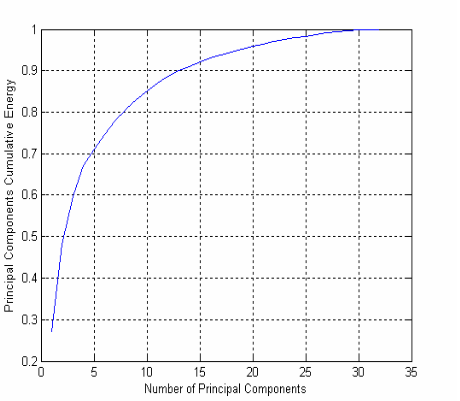


Figure 3. PCA components' cumulative energy curve

- Cosine of the angle between two vectors

S1: HSM($\mathbf{X}$,$\mathbf{MSH}$) = $\mathbf{X}^T\mathbf{MSH}/(\|\mathbf{X}\|*\|\mathbf{MSH}\|)$
where $\|.\|$ is the euclidean norm

- Histogram intersection

S2: HSM($\mathbf{X}$,$\mathbf{MSH}$) = $\sum_i (\min(\mathbf{x}_i,\mathbf{m}_i))/(\sum_i \mathbf{x}_i)$

- Bhattacharya distance measure

S3: HSM($\mathbf{X}$,$\mathbf{MSH}$) = $\sum_i (\sqrt{(\mathbf{x}_i * \mathbf{m}_i)})$

- Quadratic distance

S4: HSM($\mathbf{X}$,$\mathbf{MSH}$) = $\sqrt{}\, (\mathbf{X} - \mathbf{MSH})^T \sum^{-1} (\mathbf{X} - \mathbf{MSH})$
where $\sum$ = covariance matrix of spatial histograms for text class

The selection of similarity measure is done on the training database. The performance criterion is the discrimination power - the ratio of between-clusters-variance and within-cluster-variance. It is observed that the similarity measures S1, S2 and S3 give good results with discrimination power of 67%, 48% and 45% respectively. Quadratic distance performs worse and has a discrimination power of 36%. Histogram Intersection and Bhattacharya distance measures are designed for complete histograms, so after dimensionality reduction, we can't use them. Finally, we use S1 as a measure in feature selection due to its high discrimination power.

We want to compare the performance of this text detector with the neural classifier. So false alarm rate of neural classifier is used in this feature selection method. Table 1. shows the number of principal components selected during feature selection method.

Table 1. Number of principal components chosen by feature selection method for different window size

Window Size	5x5	7x7	9x9	11x11	13x13	17x17
Number of principal components	32	2	2	2	2	4

The text detector gives 64.3% text detection rate and 22% false alarms on window size 17x17. Although, this text detector performs slightly poorer than the neural one but it is much simpler i.e. has few parameters. Only 4 components are used along with a simple histogram similarity criterion.

The text detection performance of above detectors is shown in figure 4. There is a slight difference in performance between neural classifier and generative model. On the other hand, maximum a posteriori classifier does not perform good due to high percentage of false alarms. The influence of window size on text detection is shown in figure 5. Characters are gradually detected as window size increases. Finally, some of the text detection results are shown in figure 6.

4.4.3. Comparison of spatial histograms with GLCM features

In this section, we will compare the performance of proposed text detectors with our previous work [9].

In our earlier work, we have used 6 features namely contrast, homogeneity, dissimilarity, entropy, energy and correlation proposed by Haralick [2], for text detection task. We employed maximum likelihood classifiers assuming mono & multi gaussian distributions for text and non-text classes and a neural classifier as text detectors. The maximum likelihood classifiers are: mono gaussian for text and non-text class (TNTSG), mono gaussian for text class (TSG), two gaussians for text and non-text class (TNTMG), two-gaussians for text class (TMG). Mahalanobis distance is used to compute likelihood estimate. The neural classifier (NC) is a two layer perceptron with 6 inputs, 20 hidden cells and 2 outputs. We have observed that two class model (TNTSG or TNTMG) is better than the single class model (TSG or TMG). Moreover, mono gaussian works better than two gaussians model. The neural classifier gives the best results: text detection rate is 64% and false alarm rate is 25%. On comparing, we can see that text detectors based on spatial histograms perform better than the GLCM features ones – an increase of 2% in text detection rate and a decrease of 3% in false alarms.

Spatial histograms are fast to compute as the number of operations required is less than that associated with GLCM features. The average time to calculate GLCM features on an image of 480x640 pixels using 17x17 window is 196 seconds while it is 135 seconds for the calculation of spatial histograms.

5. Conclusions and future work

In this paper, we have employed a simple texture coding scheme for text detection in natural scene images. We observe that spatial histograms computed from GLCM are better candidates for text detection task than GLCM features. Although, the performance is evaluated on a small test database of 50 images but the results are encouraging and we hope that performance evaluation of these text detectors on a larger database will validate these results and conclusions. We have shown that a simple generative model works equally well when compared to a neural classifier and the number of histogram’s components required for effective classification is far less than the histogram dimension.

We are also working on a combination of these classifiers and hope the overall performance will improve. Currently, we have not filtered any detected text region by applying validation methods e.g. geometrical and spatial constraints, baseline detection, character alignment etc. We believe that such validation schemes will lower the false alarm rate.

Furthermore, we are exploring gradient methods as they can differentiate text and non-text regions. Gradient methods are rapid in calculation so one such method can be used to generate candidate text regions which can further be processed by our proposed texture scheme, thus making overall process fast.

6. References

[1] R. Velázquez, F. Maingreaud and Edwige E. Pissaloux, Intelligent Glasses: A New Man-Machine Interface Concept Integrating Computer Vision and Human Tactile Perception, EuroHaptics 2003, Dublin, Ireland, July 2003.

[2] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein, Textual Features for Image Classification," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp.610-621, 1973.

[3] Victor Wu, Raghavan Manmatha, Edward M. Risemann, Text Finder: An Automatic System to Detect and Recognize Text in Images, IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 21, no. 11, pp. 1224-1228, 1999.

[4] Keechul Jung, Kwang I. Kim, Takeshi Kurata, Masakastu Kourogi, Jung H. Han, Text Scanner with Text Detection Technology on Image Sequences, Proceedings of 16th International Conference on Pattern Recognition (ICPR), vol. 3, pp. 473-476, 2002.

[5] David Crandall, Sameer Antani, Rangachar Kasturi, Extraction of Special Effects Caption Text Events From Digital Video, International Journal on Document Analysis and Recognition (IJDAR), vol. 5, pp. 138-157, 2003.

[6] Julinda Gllavata, Ralph Ewerth, Bernd Freisleben, Text Detection in Images Based on Unsupervised Classification of High Frequency Wavelet Coefficients, Proceedings of 17th International Conference on Pattern Recognition (ICPR), vol. 1, pp. 425-428, 2004.

[7] Tomoyuki Saoi, Hideaki Goto, Hiraoki Kobayashi, Text Detection in Color Scene Images Based on Unsupervised Clustering of Multi-channel Wavelet Features, Proceedings of Eight International Conference on Document Analysis and Recognition (ICDAR), pp. 690-694, 2005.

[8] Jian Liang, David Doermann, Huiping Li, Camera-based Analysis of Text and Documents : A Survey; International Journal on Document Analysis and Recognition (IJDAR) vol. 7, pp. 84-104, 2005

[9] _____ , Texture based Text Detection in Natural Scene Images – A Help to Blind and Visually Impaired Persons, Conference and Workshop on Assistive Technology for People with Vision and Hearing Impairments, Euro-Assist-5, Granada, Spain, August 2007 (To appear).

[10] ICDAR 2003 Robust Reading and Text Locating Competition http://algoval.essex.ac.uk/icdar/RobustReading.html

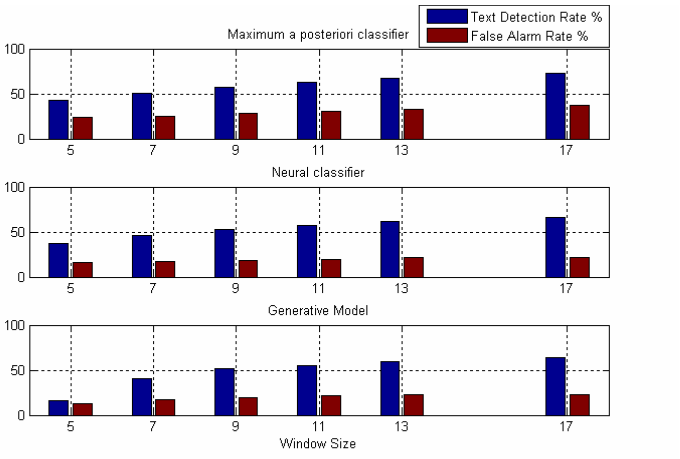


Figure 4. Performance of text detectors

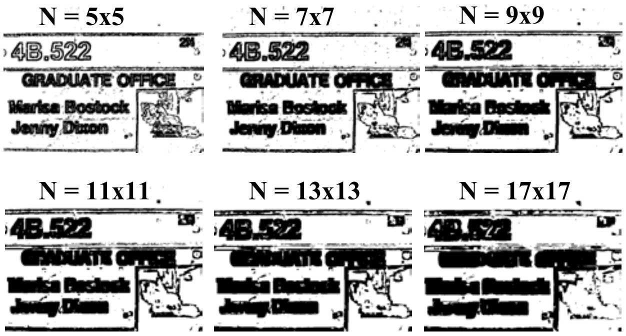


Figure 5. Effect of window size on text detection

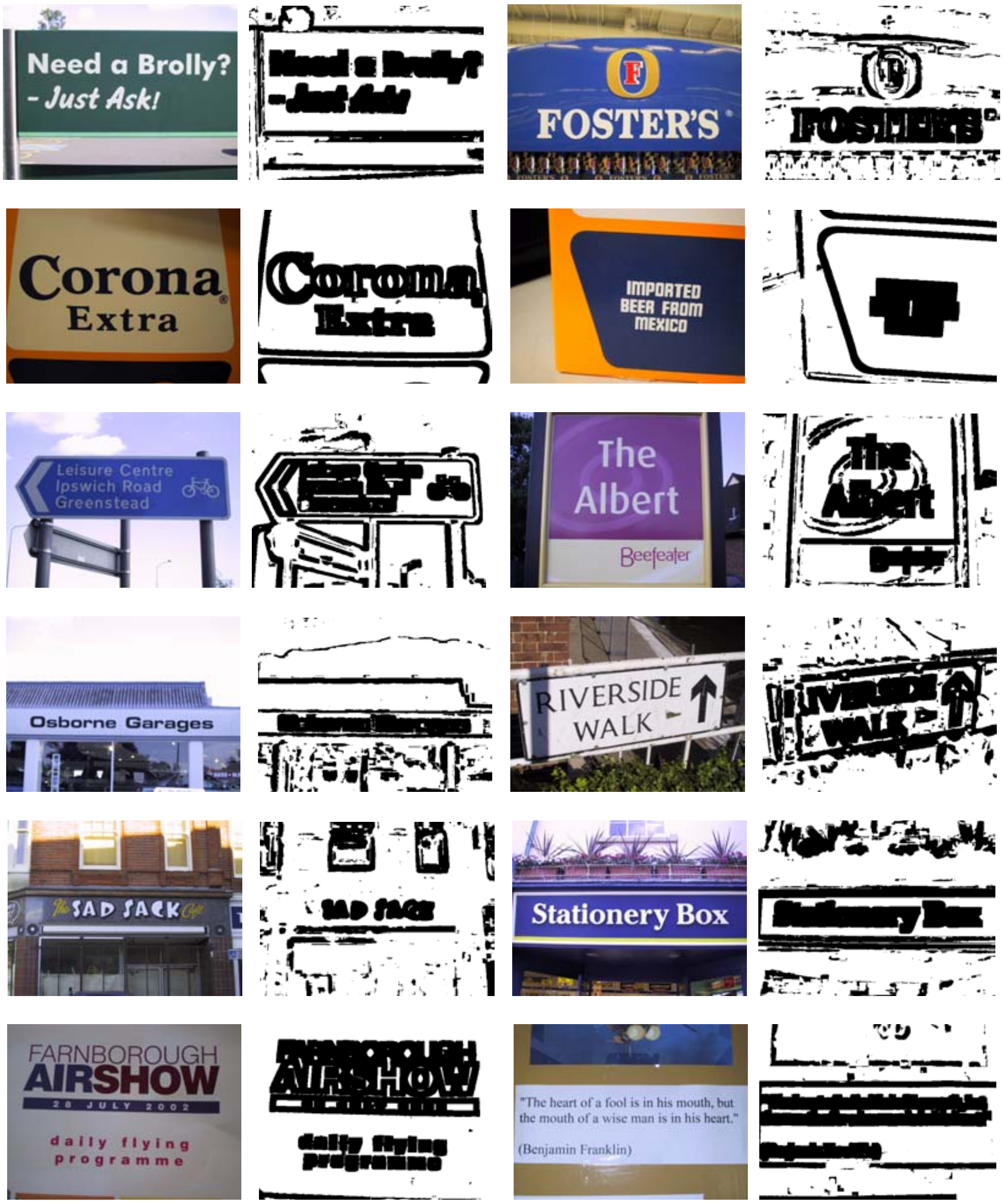


Figure 6. Text detection examples (test database)
Text detector: neural classifier with window size 17x17

Comparison between Pen-scanner and Digital Camera Acquisition for Engraved Character Recognition

Céline Mancas-Thillou
Faculté Polytechnique de Mons
Belgium
celine.thillou@fpms.ac.be

Matei Mancas
Faculté Polytechnique de Mons
Belgium
matei.mancas@fpms.ac.be

Abstract

Engraved characters are usually a limitation for character recognition, as their extraction is difficult due to the small contrast with the background. Nevertheless, they are present in several industrial applications and their handling may be convenient for special tasks such as traceability or quality control. We propose here an entire system from acquisition to character recognition by considering two different and mobile acquisition devices, a pen-scanner close to traditional acquisition and a digital camera in order to appreciate performance and differences. In some industrial applications such as the ones including engraved characters, it may be difficult and awkward to properly build a large training database for recognition. Hence, we also propose a novel method to increase automatically the size of the training dataset without loosing representativeness by using image analogies. Results are finally presented in terms of Precision and Recall for character extraction quality and recognition rates for character recognition with comparison with a commercial OCR.

1. Introduction

Machine vision and image processing algorithms have attempted for several years to be used in industrial applications. With the drastic expansion of digital cameras with reasonable prices, the particular domain of pattern recognition offers promising applications for industry. A gap needs to be filled in order to make image-based methods more efficient and versatile. One of the existing limits, nowadays, is for engraved character recognition. As stated in [13], camera-based techniques for character understanding give lower recognition rates for engraved characters. In industry, those patterns are quite present, especially on metallic objects with registration numbers, for example. Hence additionally to the engraved issue, these surfaces are reflective, which is still a topic in research in order to build robust algorithms against uneven lighting. The challenge is double: efficient understanding of engraved characters on reflective surfaces and training database constitution. Both points will be addressed in this paper.

For the first issue, a convenient mobile acquisition is needed. For several reasons of portability, robust and light devices are often required. Moreover, engraved characters are part of objects, which are very rarely flat. Hence, usual scanner-based acquisition can not be a solution. Contrarily to traditional scanners, pen-scanners offer mobility, handling of non-paper objects in every context and may constitute a real choice for industrial applications. Similarly, digital cameras give the same benefits and have been extensively used in several industrial applications such as tourist dedicated translation systems [25] or reading devices for visually impaired [14]. Video-based systems have been tested in this context and are not competitive in terms of results quality for acceptable prices. Heavy processing using super-resolution algorithms is required and leads to low-quality results compared to pen-scanners and still cameras. For these reasons, a comparison between these two devices will be addressed in the following sections.

The second issue of training database constitution for character recognition is a common problem in all pattern recognition algorithms. However, it is even more highlighted with engraved characters present in industrial situations. It is difficult to acquire large databases with various examples. No dedicated training for engraved characters lead to poor results. Hence a solution for the artificial increase of training database is absolutely needed.

Character extraction and recognition in the context of engraved character present several applications, such as quality control or traceability. Numbers need to be clearly engraved without ambiguity for end-users, especially in the case of use-before date on cans or for inspection at different creation times for risky objects, such as weapons (Figure 1). Character recognition is an interesting alternative for unalterable objects. Several other situations may also be listed.

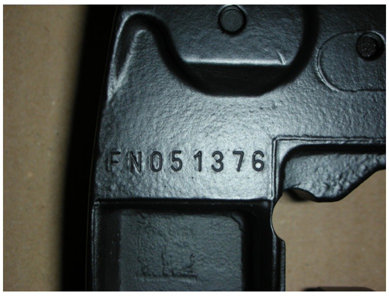


Figure 1. A registration number on a weapon butt (Copyright FN Herstal, Belgium, 2007).

Section 2 will be referred to the state-of-the-art of character extraction, useful for engraved characters. Section 3 will be presented our methods in terms of character extraction and uneven lighting removal for both mobile acquisition devices. In Section 4, our training database constitution and artificial increase will be presented with our inhouse character recognition. Section 5 will be addressed to detailed results in terms of character extraction and recognition. Comparisons with existing algorithms will also be mentioned. Finally, in Section 6, a discussion will conclude this paper along with presentation of our future works.

2. State-of-the-Art

As far as we know, no systems have been dedicated to engraved or embossed characters, mainly due to the recent work on camera-based systems and their analysis complexity. Recent color-based text extraction can not efficiently use color information [13] for this kind of characters, present in a metallic environment. The main point is the close similarity between color foreground and background. Moreover, due to the metallic environment, colors are close to the main RGB diagonal, meaning they mostly represent variations of gray. Hence, these methods compete with binarization systems from gray-level images.

Natural scene character extraction algorithms are classified into several categories, from the simplest ones to the most difficult ones:
Thresholding-based methods define a threshold globally (for the whole image) or locally (for some given regions) to separate text from background. Histogram-based thresholding is one of the most widely used techniques for monochrome image segmentation. The threshold is chosen as the value corresponding to the valley between two peaks. The most referenced method is the one described by Otsu [17] and used for a visually impaired-driven application in [3, 21] . These methods work well with low computational resources and are applied mostly on gray-scale images. Adaptive or local binarization techniques define several thresholds $T(i, j)$ for different image parts depending upon the local image characteristics. Several papers [11, 24] for video text extraction used the Niblack's method [16] where the threshold depends on local mean and standard deviation over a square window of size to define. An extension is the method of Sauvola and Pietikäinen [19]. This adaptive technique is in use in *Mobile Reader^{TM}* [9], a mobile phone reading text from Inzisoft. Adaptive binarizations may handle more degradations (uneven lighting, varying colors) than global ones but suffer to be too parametric which is not versatile. Moreover, these techniques still consider gray-scale images only and were mainly used for video caption text with clean backgrounds.

Entropy-based methods use the entropy of the gray levels distribution in a scene. Li and Doermann [11] minimized the cross-entropy between the input video gray-scale frame and the output binary image. The maximization of the entropy in the thresholded image means that a maximum of information was transferred. Du et al. [2] compared Otsu's binarization and different entropy-based methods such as Pal and Pal [18]'s local entropy, joint entropy and the joint relative entropy which performs best on RGB channels independently for video caption text. Entropy-based techniques have been little referenced in natural scene (NS) context and applied only on gray-scale images or separate channels of a particular color space.

Clustering-based approaches group color pixels into several classes assuming that colors tend to form clusters in the chosen color space. Clustering-based algorithms are the most renowned and efficient methods for scene images. They are often considered as the multidimensional extension of thresholding methods. Nevertheless, in NS analysis, colors are mostly used in different color spaces to handle color properties. However, in the context of engraved characters where gray-levels values are more appropriate to use, clustering-based methods may be more accurately defined by the gray-level clustering into two parts as background and foreground (characters). The most popular method is k-means, which aims at minimizing an objective function, which is the sum-of-squared error criterion, to build representative clusters, meaning that points inside a cluster are more similar than those inside another cluster. Its generalization, Gaussian Mixture Modeling (GMM), is more and more exploited. K-means clustering in NS text extraction has been extensively used under various forms, either performed in the RGB color space [10], in HSI [22], in YCbCr [4] or in a dynamically uncorrelated color space using principal components analysis [1]. As main drawbacks, clustering methods suffer from the need to previously set up the number of clusters and initialization variation leading to different segmentations. Problems of initialization are traditionally solved by multiple computations based on random initialization to reduce this effect towards convergent results. For the number of clusters to set, it is either prefixed or dynamically computed, with 3D histogram analysis in [10], for example.

Reviews of binarization/text extraction methods have already been done, hence the reader is referred to one of the excellent surveys, the one of Sankur and Sezgin [18]. Related to this survey, an educational software has been delivered and have been tested for results comparison.

About pen-scanners images, similar algorithms may be applied. Usually pen-scanners come with off-the-shelf optical character recognition (OCR) from various companies. However, even if degradations are less numerous in pen-scanner acquisition than in camera-based one, these commercial OCRs fail faced to engraved characters, mainly due to low contrast, dirtiness in engravings and so on. In the following section, we will present our system to handle engraved character extraction on reflective surfaces for both acquisitions.

Industry-driven systems present an additional issue, which is the building of the training database for recognition. A few samples are available for analysis in a short given period. Hence, text database for a particular application is awkward and difficult to achieve. To circumvent this effect, Section 4 will deal with this problem. However, some works have been done related to the training database increase. Traditional database increasers are based on geometrical deformations such as affine transformations or on the reproduction of a degradation model such as [20] to mimic NS issues. Other alternatives using ensembles of classifiers based on either a unique and extended dataset or different recognizers or by adding multiple preprocessing are also possible. Nevertheless, results are rarely as good as the use of a dedicated database for an exotic application, hence we chose the first solution and we automatically built a large training database.

3. From Acquisition to Character Extraction

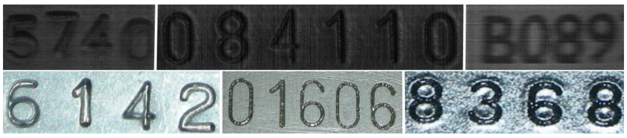


Figure 2. Samples of acquisition. Top: using a pen-scanner, bottom: using a digital camera.

Even if pen-scanner based acquisition may seem to overpass results done with a digital camera, intensive works have been done on camera based acquisition during these previous years and it is interesting to compare both performances in order to have real choices for industrial applications.

3.1. Mobile acquisition

Acquisition done with a mobile pen-scanner is obviously easy, quite intuitive for non-expert people. The idea is to scan the characters to be recognized by an horizontal motion (for Latin words). Stabilization of pen-scanners is quite robust; hence, even if translation speed is not uniform, the image has a good quality as shown in Figure 2. However, some points may be real obstacles for some applications, such as the low contrast for engraved characters because lighting induced by the pen is quite low. Additionally, a flat surface of about 2-3 cm is needed to properly acquire characters. If this assumption is not checked, the acquisition may end without having acquired all characters. Hence, it may be not the case when the surface is curved or characters close to another object. Regarding this point, some characters may be very difficult to take as they need to be in contact with the scanner, which may difficult in some applications. For example, on a line for quality control of cans, a pen-scanner needs intervention of a user when digital camera may be turned ‘ON’ automatically. Digital cameras are also mobile devices with the main benefit for use that everybody (or almost) has ever used one. Connection with a computer may be direct or not and prices are now equivalent to those of pen-scanners. All issues of pen-scanners are avoided with an acquisition through a still camera. Hence, it is more versatile. Nevertheless, main disadvantages are due to inherent degradations of natural scenes, such as uneven lighting, blur, surface reflections or perspective. Usually, in natural scenes, more problems may be cited such as complex background, large diversity of fonts and sizes but in the case of industrial applications, these points do not exist. Samples are shown in Figure 2. Finally, the additional step needed for camera-based images compared to those of pen-scanners is the text localization. Several researchers [23, 26] have been working on this step and in a constraint environment with simple background, it is easier. It will not be mentioned in this paper as we compare text extraction and recognition for engraved characters and we assume textual areas are available. Nevertheless, for some trials, some text locaters are available online[1] and issued from the ICDAR 2003 Competition.

3.2. Engraved character extraction

Due to difference between number of degradations for both acquisition, either pen-scanner or digital camera, the

[1] http://algoval.essex.ac.uk:8080/textloc/

character extraction is obviously not the same. Pen-scanner based images need only a global processing, an Otsu grayscale thresholding [17], as assessed in Section 5 while camera-based ones require dedicated processing. Different analyzes are described in Figure 3.

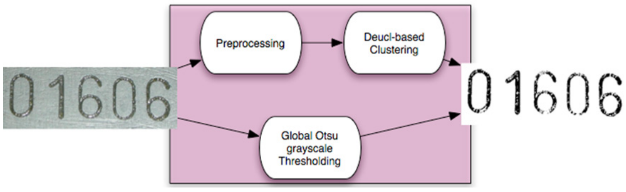


Figure 3. Different character extraction for both acquisition types: top: using a still camera, bottom: using a pen-scanner.

To circumvent lighting and blur effects, which are the main damageable degradations for camera-based images I, the dedicated character extraction is composed of a preprocessing and a clustering-based segmentation. A contrast enhancement [12] is applied as a pre-processing, which is issued from visual system properties and more particularly on retina features and leads to $I_{enhanced}$:

$$I_{enhanced} = I * H_{gangON} - (I * H_{gangOFF}) * H_{amac} \quad (1)$$

with

$$H_{gangON} = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & 2 & 3 & 2 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

$$H_{gangOFF} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -2 & -1 & 1 \\ 1 & -2 & -4 & -2 & 1 \\ 1 & -1 & -2 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} H_{amac} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

These three previous filters assess eye retina behavior and correspond to the action of ON and OFF ganglion cells $(H_{gangON}, H_{gangOFF})$ and of the retina amacrine cells (H_{amac}). The output is a band-pass contrast enhancement filter which is more robust to noise than most of the simple enhancement filters. Meaningful structures within the images are better enhanced than by using classical high-pass filtering which provides more flexibility to this method. Afterwards the information from this enhancement technique may be integrated in order to quantify the interest of some regions in an image [12], but we only use here the image enhancement results.

Following this contrast enhancement, a median filtering is applied to remove texture of metal and spurious parts of engraving and leads to $I^m_{enhanced}$, as shown in Figure 4.

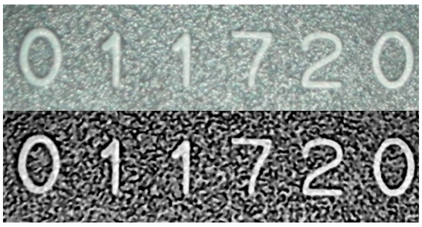


Figure 4. Result (bottom) of our preprocessing applied on an original image (top).

Based on this robust contrast enhancement, a clustering-based segmentation $\mathcal{C}$ is then applied, leading to $I_{binarized}$:

$$I_{binarized} = \mathcal{C}(I^m_{enhanced}) \quad (2)$$

We exploit color information to handle varying colors inside textual areas, especially those induced by uneven lighting or flash effect when needed. In order to segment similar colors together, we use an unsupervised segmentation algorithm with a fixed number of clusters. In this paper, the focus is done on how natural scene text can be extracted to increase recognition results; we consider here only already detected text areas. As areas are constrained, we use a 3-means clustering. The identification of clusters is a textual foreground, a background and a noisy cluster which consists either in noise in badly illuminated images or in edges of characters, which are always slightly different, in clear images.

First, a color reduction is applied. Considering properties of human vision, there is a large amount of redundancy in the 24-bit RGB representation of color images. We decided to represent each of the RGB channels with only 4 bits, which introduce very few perceptible visual degradation. Hence the dimensionality of the color space C is $16 \times 16 \times 16$ and it represents the maximum number of colors.

Following this initial step, we use the 3-means clustering to segment C into three colored regions. The three dominant colors (C_1, C_2, C_3) are extracted based on the centroid value of each cluster. Finally, each pixel in the image receives the value of one of these colors depending on the cluster it has been assigned to. Among the three clusters, one represents obviously background. The background color is selected very easily and efficiently as being the color with the biggest rate of occurrences on the image borders. Only two pictures left which correspond depending on the initial image to either two foreground pictures or one foreground picture and one noise picture.

A new measure M is introduced to find the most textual foreground cluster over the two remaining clusters. Based on properties of connected components of each cluster, spa-

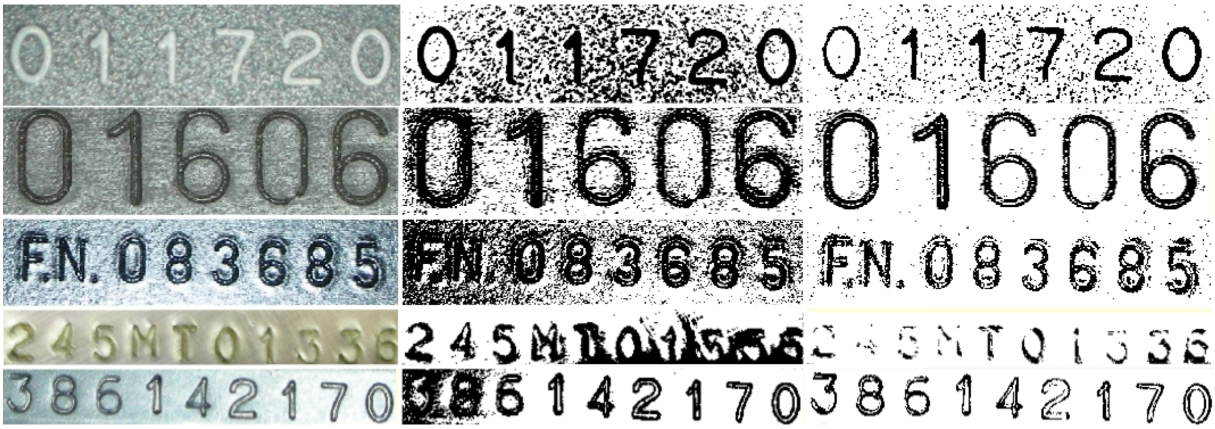


Figure 5. Character extraction results. 1st column: original images, 2nd column: Otsu thresholding [17], 3rd column: our human vision-based pre-processing followed by 3-means clustering.

tial information is added at this point to find the main textual cluster. M is based on a larger regularity of connected components of text than the one of noise and background and is defined as below:

$$M = \sum_{i}^{N} |area_i - \frac{1}{N}(\sum_{i}^{N} area_i)| \qquad (3)$$

where N is the number of connected components and $area_i$, the area of the component i. This measure enables to compute the variation in candidate areas. The main textual cluster is identified as the one having the smallest M. If the last cluster belongs to text, both clusters need to be merged. A new computation of M is done considering the merging of both clusters. If M decreases, the merge is processed.

Some results are displayed in Figure 5 where comparison is done with a classical global thresholding, method which performs better on constrained grayscale areas [13]. Additionally, a connected component analysis is then performed and small ones are removed. Moreover, our solution, presented in Section 4, enables to allow for such degradations.

Some images still fail due to absolute luminance variation, meaning that textual areas are separated into two clusters in very badly illuminated images and sometimes also due to curved surfaces. Actually, characters are darker than background in some partial areas of the image and inversely in some other parts, as shown in Figure 6. Local thresholding-based extraction do not perform better for this kind of images and other pre-processings such as uneven lighting removal [5] have also been tested without success. Nevertheless image analogies, used in Section 4, may be a smart solution to this issue or the use of region-growing methods by taking into account the "engraved" property of characters.

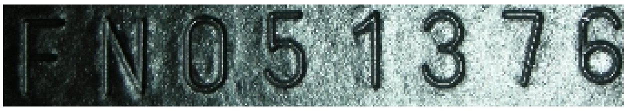


Figure 6. Sample of badly illuminated images with absolute luminance variation.

4. Training Database Constitution and Character Recognition

In order to get better results, supervised classification is traditionally used for character recognition and needs representative and large training database, which is sometimes difficult to build properly. In our system, we choose in-house character recognition, which will be described in Subsection 4.2 and propose an innovative algorithm for constitution of the training database. This novel method enables to support pattern recognition for particular applications, such as the one of engraved or embossed characters, present in several industrial contexts.

4.1. Artificial increase of databases

Our training database increase algorithm is based on the image analogies of Hertzmann et al. [8], with the particular method of texture-by-numbers.

Given a pair of images A and A', with A' being the binarized version of A, the textured image in our algorithm, and B' the black and white image to transfer texture, the texture-by-numbers technique applies texture of A into B' to create B. Binary versions are composed of pixels having values of 0 or 1; texture of A corresponding to areas of 0 of A' will be transferred to areas of 0 of B' and similarly for 1. Multiscale representations through Gaussian pyramids are computed for A, A' and B' and at each level, statistics for every pixel in the target pair (B, B') are compared to every pixel in the source pair (A, A') and the best match is found. The number of resolution levels and the neighborhood size to find the best similarity is previously defined. The similarity is based on the traditional Euclidean distance and the neighborhood search to find the best pixels for texture transfer is based on approximate nearest neighborhood (ANN) and tree-structured vector quantization (TSVQ). Additional mathematical information may be found in [7]. The result of texture transfer is displayed in Figure 7.

Hence the main idea is to create several binary characters (B') and to apply texture of a small set of natural scenes images (A) upon these characters. The binary versions of A are computed with the character extraction of Section 3. We then get a large dataset of automatically rendered characters (B) with all degradations of natural scenes. This new and artificially created database may be then binarized and obviously leads to a very representative database for future tested characters.

We first need to build several character-templates to enlarge the training database and to apply the texture-by-numbers method. Based on a given set with a particular font, lines and curves of characters are modeled with cubic splines thanks to five anchors per character maximum. To build templates as various and realistic as possible, several parameters may be then defined to add degradations based on the defined anchors. Variations induce different global and local thicknesses, perspective, rotation of characters or modify the contour (squared or curved) or the ends with some artistic display such as shown in Figure 7.

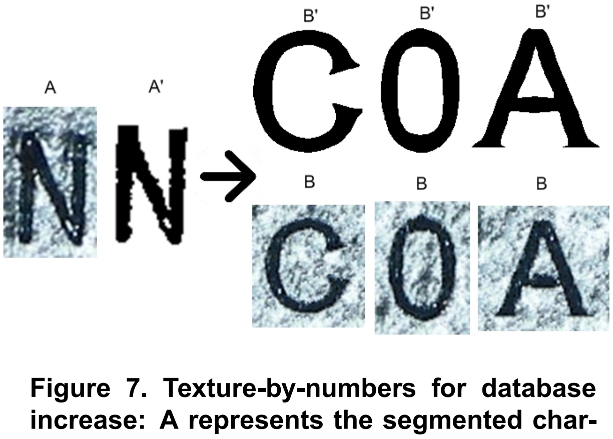


Figure 7. Texture-by-numbers for database increase: A represents the segmented character, A' its binary version, B' are character-templates for texture to transfer onto B.

Based on the finite steps of variation for each of the precited parameters, for one extracted character and one given texture, 33480 samples may be created. Hence, the power of increasing database of this method is very large (almost infinite depending on the parameter variation and the number of textures).

One limitation of this method is when no representative contours are available in the A samples. For example, if A represents a '0' and we wish to transfer a texture upon the character 'N' with no curved contours, similar as '0'. Moreover '0' has no strict lines, similar as 'N'. Hence, to perform realistic transfers it is needed to use several A samples in order to build representative patches to apply the texture-by-numbers algorithm and we build new textured characters with several A samples ('0', '2', '5', 'A', 'B', 'C', 'E', 'N').

4.2. In-house character recognition

We use an extended version of classifier from Gosselin [6], based on geometrical features and a multi-layer perceptron (MLP). In order to recognize many variations of the same character, features need to be robust against noise, distortions, translation, rotation or shear. Invariants are features which have approximately the same value for samples of the same character, deformed or not. To be as invariant as possible, our input-characters are normalized into an N*N size with N=16. However, not all variations among characters such as noise or degradations can be modeled by invariants, and the database used to train the neural network must have different variations of a same character, with representativeness of issues. The previous algorithm of training database increase fulfills this requirement.

In our experiments, we use a feature extraction based on contour profiles. The feature vector is based on the edges of characters and a probe is sent in each direction (horizontal, vertical and diagonal) and to get the information of holes like in the 'B' character, some interior probes are sent from the center. Moreover, another feature is added: the ratio between original height and original width in order to very easily discriminate an 'i' from an 'm'. Experimentally, in order to lead to high recognition rates, we complete this feature set with Tchebychev moments, which are orthogonal moments. Moment functions of a 2D image are used as descriptors of shape. They are invariant with respect to scale, translation and rotation. According to Mukundan et al. [15], we use Tchebychev moments of order 2 for their robustness to noise.

No feature selection is defined and the feature set is a vector of 63 values provided to an MLP with one hidden layer of 120 neurons and an output layer of variable size, depending on applications (36 for Latin letters and digits or 10 for digits only, for example). The total number of training samples is divided into 80% for training only and 20%

for cross-validation purpose in order to avoid overtraining.

5. Results

To assess quality of our proposition of algorithms, results are given for character extraction and recognition independently and are based on a database of natural scene images, acquired *in situ* either with a pen-scanner ("Iris pen executive") or a digital camera (with a resolution of 4 Megapixels). A small database of 30 images is available. As stated previously, large databases may be difficult to get. However, this database has been carefully built with several types of reflective surfaces (anodized, phosphated, etc) with characters engraved at different places in the object.

For character extraction, some comparisons done with the global Otsu thresholding [17] have already been displayed in Figure 5 but other algorithms in the domain of gray-scale character binarization have been tested. Sankur and Sezgin [18] implemented a comparison software OTIMEC of 38 extraction methods (version 4.1) and we ran it to compare results with our proposition. The description of all algorithms is out of focus of this paper and the reader may refer to their excellent survey [18].

Results are given in terms of Precision and Recall. Precision measures the quality of extraction while Recall measures the quantity of high quality extraction. "Correctly extracted characters" means characters which are extracted without noise or missing important parts of the character. When differences between methods are small (a few negligible pixels), identical rates are considered. Most differences are (very) large with the absence of character parts or even image areas due to uneven illumination. Hence visual assessment is easy to handle and not damageable for results.

$$\text{Precision} = \frac{\text{Correctly extracted characters}}{\text{Total extracted characters}} \qquad (4)$$

$$\text{Recall} = \frac{\text{Correctly extracted characters}}{\text{Total number of characters}} \qquad (5)$$

Best results are obtained for the Otsu thresholding in the case of pen-scanner images with a Precision of 0.74 and a Recall of 0.70. For still camera-based images, our proposition outperforms the 38 algorithms of OTIMEC and we got a Precision of 0.83 and a Recall of 0.74.

To give figures in terms of recognition rates, we compare with a commercial OCR (ABBYY FineReader 8.0 Professional Edition Try&Buy [2]) in order to apprehend also image quality, which is not obvious with some thumbnails samples only. Results are given in Table 1. Recognition rates are not sufficient for an industrial application. Nevertheless, a correction of recognition errors may be applied regarding a particular application. Moreover, acquisition may still be more accurately tuned to get better results. Synthesized data slightly improved recognition rates (around 2%) when compared with a generic NS training database. Nevertheless, image analogies enable an easy building of large databases (if none is available) and give also a strong solution to imperfect text extraction by embedding usual degradations. One point is important to mention: the comparison of recognition rates between both acquisition. The pen-scanner based one is better by its simplicity and uniform quality of images even if the contrast may still be poor. Nevertheless, the difference between recognition rates is not very large and camera-based acquisition may now be considered as a competitive alternative acquisition mode, even for industrial applications.

Table 1. Recognition rates for engraved characters with comparison with a commercial OCR, either for pen-scanner (PS) or digital camera (DC) acquisition.

	Comm. OCR	Our method
PS	26.92%	80.77%
DC	21.13%	76.06%

6. Conclusion and Discussion

In this paper, we presented a character recognition system for engraved characters on reflective surfaces by considering two different acquisition devices. The main aim was to compare results in order to understand the gap between these both acquisition devices. Degradations of camera-based images are more numerous and the one leading to more errors is uneven lighting with absolute luminance variation. We proposed two different character extractions for both acquisition modes and assessed the performance of these methods by comparing them with other grayscale thresolding methods. In order to build an efficient recognizer, the quality and representativeness of the training database is fundamental. For some industrial applications, which is often the case for engraved characters, it is sometimes difficult to get large datasets. Hence, we presented a novel method for training database increase based on character templates constitution through cubic splines and image analogies with the texture-by-numbers algorithm. Results are satisfying and degradations of natural scene images are well rendered, enabling the building of a large and realistic training database. This method is versatile and may

[2] http://france.abbyy.com/download/?param=46440

be applied for any kinds of training database to be built and is very convenient for all sets of natural scene images or for applications requiring various datasets. Results for camera-based acquisition have to be mitigated with performance of the text detection step, which is, according to us, not a real problem but which may slightly lower results. As the difference between recognition rates is not very large, camera-based acquisition may now be considered as an interesting alternative for industrial applications and especially when requirements of pen-scanner acquisition are not met. Several future works may be formulated. Tests on a larger database for character extraction are needed among different industrial applications. Moreover, image analogies applied in the context we described may be used for several other steps of image processing such as uneven lighting removal, thresholding and so on.

7. Acknowledgement

Our acknowledgements are directed towards the Walloon Region, which financed this project under a post-doctoral contract. We address special thanks to A. Hertzmann and the co-authors for their excellent work on image analogies. Finally, we wish to thank Levente Jeney, who partially contributed to some sections of this paper during his Master studies.

References

[1] F. Drira and H. Emptoz. A recursive approach for bleed-through removal. In *Proc. Camera-based Doc. Analysis and Recognition*, pages 119–126, 2005.

[2] Y. Du, C.-I. Chang, and P. Thouin. Unsupervised approach to color video thresholding. *Optical Engineering*, 43(2):282–289, 2004.

[3] N. Esaki, M. Bulacu, and L. Shomaker. Text detection from natural scene images: towards a system for visually impaired persons. In *Int. Conf. on Pattern Recognition*, pages 683–686, 2004.

[4] L. Fu, W. Wang, and Y. Zhan. A robust text segmentation approach in complex background based on multiple constraints. In *Proc. Pacific Rim Conf. on Multimedia*, pages 594–605, 2005.

[5] B. Funt, F. Ciurea, and J. McCann. Retinex in matlab. In *Proc. of the IS&T Color Imaging Conference: Color Science, Systems and Applications*, pages 112–121, 2000.

[6] B. Gosselin. *Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits*. PhD thesis, Faculté Polytechnique de Mons, 1996.

[7] A. Hertzmann. *Algorithms for rendering in artistic styles*. PhD thesis, New York University, 2001.

[8] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin. Image analogies. In *Proc. ACM SIGGRAPH, Int. Conf. On Computer Graphics and Interactive Techniques*, 2001.

[9] I.-J. Kim. Multi-window binarization of camera image for document recognition. In *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, pages 323–327, 2004.

[10] J. Kim, S. Park, and S. Kim. Text locating from natural scene images using image intensities. In *Int. Conf. on Doc. Analysis and Recognition*, pages 655–659, 2005.

[11] H. Li and D. Doermann. Text enhancement in digital video using multiple frame integration. In *Proc. ACM Int. Conf. on Multimedia*, pages 19–22, 1999.

[12] M. Mancas, C. Mancas-Thillou, B. Gosselin, and B. Macq. A rarity-based visual attention map - application to texture description -. In *Proc. of IEEE Int. Conf. on Image Processing*, 2006.

[13] C. Mancas-Thillou. *Natural Scene Text Understanding*. PhD thesis, Faculté Polytechnique de Mons, 2006.

[14] C. Mancas-Thillou, S. Ferreira, J. Demeyer, C. Minetti, and B. Gosselin. A multifunctional reading assistant for the visually impaired. *EURASIP Int. Jour. on Image and Video Processing*, To appear, 2007.

[15] R. Mukundan, S. Ong, and P. Lee. Discrete vs. continuous orthogonal moments in image analysis. In *Proc. of Int. Conf. On Imaging Systems, Science and Technology*, pages 23–29, 2001.

[16] W. Niblack. *An introduction to image processing*. Prentice-Hall, 1986.

[17] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. System, Man and Cybernetics*, 9(1):62–66, 1979.

[18] B. Sankur and M. Sezgin. A survey over image thresholdng techniques and quantitative performance evaluation. *Jour. Electronic Imaging*, 13(1):146–165, 2004.

[19] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.

[20] J. Sun, Y. Hotta, and Y. Katsuyama. Low resolution character recognition by dual eigenspace and synthetic degraded patterns. In *Proc. ACM Hardcopy Doc. Processing Workshop*, pages 15–22, 2004.

[21] C. Thillou, S. Ferreira, and B. Gosselin. An embedded application for degraded text recognition. *Eurasip Jour. on Applied Signal Processing, Special Issue on Advances in Intelligent Vision Systems: methods and applications*, 13:2127–2135, 2005.

[22] K. Wang and J. Kangas. Character location in scene images from digital camera. *Pattern Recognition*, 36:2287–2299, 2003.

[23] V. Wu, R. Manmatha, and E. Riseman. Textfinder: an automatic system to detect and recognize text in images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(11):1224–1229, 1999.

[24] A. Zandifar, R. Duraiswami, and L. Davis. A video-based framework for the analysis of presentations/posters. *Int. Journal on Doc. Analysis and Recognition*, 7(2–3):178–187, 2005.

[25] J. Zhang, X. Chen, A. Hanneman, J. Yang, and A. Waibel. A robust approach for recognition of text embedded in natural scenes. In *Int. Conf. on Pattern Recognition*, 2002.

[26] Y. Zhong, K. Karuand, and A. Jain. Locating text in complex color images. *Pattern Recognition*, 28(10):1523–1535, 1995.

Rectifying Perspective Distortion into Affine Distortion Using Variants and Invariants

Masakazu Iwamura
Osaka Pref. Univ., Japan
masa@cs.osakafu-u.ac.jp

Ryo Niwa
Osaka Pref. Univ., Japan
niwa@m.cs.osakafu-u.ac.jp

Koichi Kise
Osaka Pref. Univ., Japan
kise@cs.osakafu-u.ac.jp

Seiichi Uchida
Kyushu Univ., Japan
uchida@is.kyushu-u.ac.jp

Shinichiro Omachi
Tohoku Univ., Japan
machi@ecei.tohoku.ac.jp

Abstract

For user convenience, document image processing captured with a digital camera instead of a scanner has been researched. However, existing methods of document image processing are not usable for a perspective document image captured by a digital camera because most of them are designed for the one captured by a scanner. Thus, we have to rectify the perspective of the document image and obtain the frontal image as if it was captured by a scanner. In this paper, for eliminating perspective distortion from a planar paper without any prior knowledge, we propose a new rectification method of a document image introducing variants *which change according to the gradient of the paper and* invariants *which do not change against it. Since the proposed method does not use strong assumptions, it is widely applicable to many document images unlike other methods. We confirmed the proposed method rectifies a document image suffering from perspective distortion and acquires the one with affine distortion.*

1. Introduction

Recently, camera-based document analysis and character recognition have been researched [3, 6]. Camera-based approach is known to be more difficult than scanner-based approach since a captured image can be degraded by nonuniform lighting, out of focus, perspective distortion and so on. Despite the difficulty, the camera-based approach has advantages of portability and ease to use. For example, scanners are not easy to carry, not able to scan a very big poster and an immovable object[1]. To the contrary, cameras are suitable to capture an image easily and quickly. This can bring us a new application of document analysis and character recognition.

Though camera-based approach has a chance to yield excellent applications, the realization is not easy. A reason is that most existing document analysis techniques are for scanner-captured images. That is, processing a camera-captured image preliminarily requires a rectification of image to obtain a scanner-captured like image. There exists many rectification methods, however, they require strong restriction on layout and way of capturing. For example, they are not applicable to the document image shown in Fig. 1. Thus. in this paper, we propose a novel rectifying method of perspective distortion of a document image. The proposed method estimates relative depth of each region of a document without any prior knowledge by employing an area of a character as a *variant* and an area ratio as an *invariant*. Since the proposed method does not use strong restriction on layout and way of capturing, it is applicable to document images of wide variety including the document image shown in Fig. 1. The comparison with existing methods is discussed in Sec. 4.

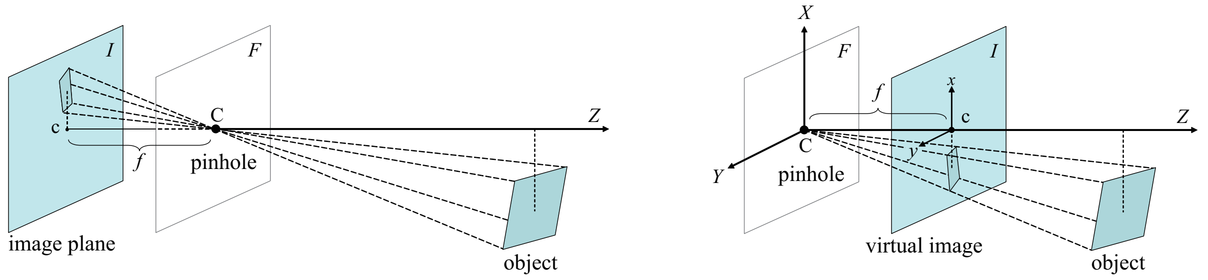
2. Proposed method

In this section, we explain the proposed rectifying method of perspective distortion. For the sake of the explanation, we begin with explaining the relationship between 3D and 2D coordinate systems.

2.1. Central perspective projection model [1, 7]

To begin with, we mention how a 2D image of a 3D object is obtained with a camera. As shown in Fig. 2(a), the

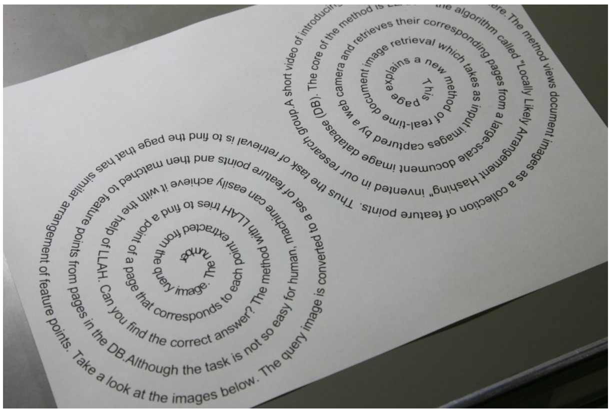
[1] A portable scanner can be easily carried. However, the size of scannable paper is constrained.



(a) The pinhole imaging model.

(b) The pinhole imaging model using virtual image. Virtual image shown in (b) is not inverted.

Figure 2. Central perspective projection model.



This page explains a new method of real-time document image retrieval which takes as input images captured by a web camera and retrieves their corresponding pages from a large-scale document image database (DB). The core of the method is the algorithm called "Locally Likely Arrangement Hashing" invented in our research group.A short video of introducin ere. The method views document images as a collection of feature points. Thus the task of retrieval is to find the page that has similar arrangement of feature points. Take a look at the images below. The query image is converted to a set of feature points and then matched to feature points from pages in the DB.Although the task is not so easy for human, machine can easily achieve it with the help of LLAH. Can you find the correct answer? The method with LLAH tries to find a point of a page that corresponds to each point extracted from the query image. The number

Figure 1. A document with difficulty to rectify for existing methods.

pinhole imaging model is often used. A point C is a *pinhole* at the *optical center*. Light rays passing through the pinhole C form an image of an object on the *image plane* I at a distance f from the optical center. An axis which passes through the pinhole C and is perpendicular to the image plane is called *optical axis*. Let c be the image center which is the intersection point between the image plane and the optical axis. In this model, parallel lines do not always be transformed into parallel lines. This transformation is called *perspective projection*. The distortion caused by the perspective projection is called *perspective distortion*.

In general, the image plane is rearranged as shown in Fig. 2(b). The image coordinate system is a 2D coordinate system which employs the image center c as the origin, and x- and y-axes as shown in Fig. 2(b). The camera-centered image coordinate system is the coordinate system which employs the focal point C as the origin, the optical axis as Z-axes, X- and Y-axes as x- and y-axes of the image coordinate system. A point $(X, Y, Z)^T$ in the camera-centered image coordinate system is projected into the point $(x, y)^T$ in the 2D image coordinate system. The transformation is written as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ Y \end{pmatrix}. \tag{1}$$

2.2. Relationship between area and depth

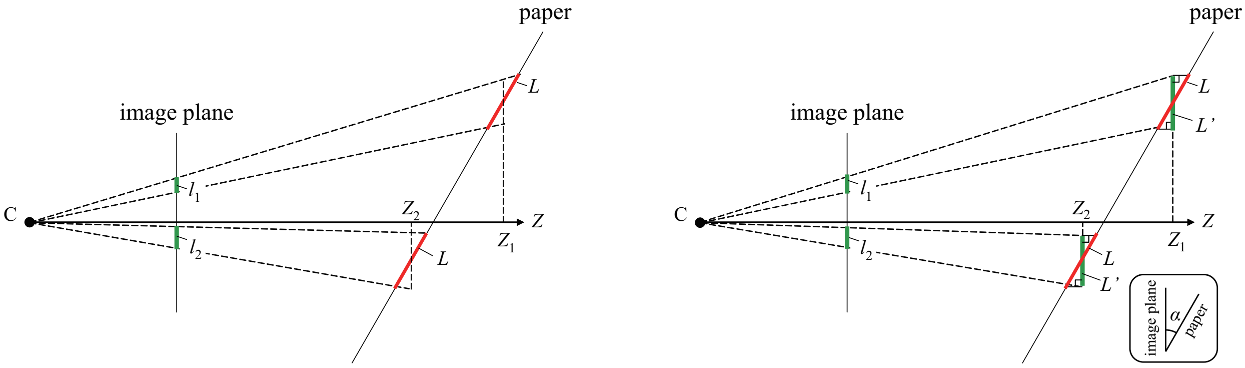
Imagine a document image suffering from perspective distortion, slanted in a certain angle. There usually exists the same characters, say "a," printed in the same size. Due to the perspective distortion, the observed sizes of the characters vary by their positions; a character near the camera is large and that far from the camera is small. We estimate the slant angle from the changes of character sizes.

Here, we derive the relationship between the observed area and depth of a character[2]. Fig. 3(a) illustrates two same characters in different positions on a document. The Z coordinates of them in the center are Z_1 and Z_2, respectively. Say, $Z_1 > Z_2$.

First of all, we consider a simplified problem handling a *length* (e.g., height or width) of a character instead of an area of it. Let L be the inherent length of the character. Let l_1 and l_2 be the projection lengths of the characters in the positions of Z_1 and Z_2, respectively. To make the following calculation easier, we use an approximation as shown in Fig. 3(b). The approximation makes the slant characters standing. Let L' be their approximated length determined as

$$L' = L \cos\alpha, \tag{2}$$

[2] Note that the area of a character means the number of pixels in foreground colors. Actually, "area of a connected component" may be more precise expression since there exists separated characters such as "i" and "j".



(a) Projection of characters.

(b) Projection of standing characters of approximated length L'.

Figure 3. The relationship between the inherent length L and the projection lengths l_1 and l_2.

where $\alpha, \ 0 \leq \alpha \leq \pi/2$, is the angle between the paper and the image plane. Thus, by considering only x coordinate in Eq. (1), we obtain

$$l_j = \frac{f}{Z_j} L' = \frac{f}{Z_j} L \cos\alpha. \tag{3}$$

Then, we consider the area of a character. Let S and S' be the area and the approximated area of a character, which correspond to L and L'. The relationship of them is given as

$$S' = S \cos\alpha. \tag{4}$$

Then, we obtain the projection area of a character s_j as follows.

$$s_j = \left(\frac{f}{Z_j}\right)^2 S' = \left(\frac{f}{Z_j}\right)^2 S \cos\alpha. \tag{5}$$

Eq. (5) shows the relationship between an observed area and depth; a projection area is inverse proportional to the square of the Z coordinate (i.e., depth). Thus, letting Z_j and s_j be the depth and observed area of the j-th character, then we obtain the following relationship from Eq. (5):

$$Z_j = \frac{f\sqrt{S\cos\alpha}}{\sqrt{s_j}}. \tag{6}$$

Finally, we consider how to determine the angle α. Since all the characters are on a coplanar in a 3D coordinate system, the angle α will be obtained by fitting them to a coplanar. The detail of the process is discussed in Sec. 2.4. Here we mention the way to calculate a 3D coordinate from a 2D image coordinate. From Eq. (1), the coordinate of the j-th character $(X_j, Y_j, Z_j)^T$ in the camera-centered coordinate system is denoted as

$$\begin{pmatrix} X_j \\ Y_j \\ Z_j \end{pmatrix} = \begin{pmatrix} Z_j x_j / f \\ Z_j y_j / f \\ Z_j \end{pmatrix}. \tag{7}$$

For the simplicity, we use the following expression instead:

$$\begin{pmatrix} X_j \\ Y_j \\ Z'_j \end{pmatrix} = \begin{pmatrix} Z'_j x_j \\ Z'_j y_j \\ Z'_j \end{pmatrix}, \tag{8}$$

where $Z' = Z/f$.

2.3. Clustering using area ratio

The method to estimate the slant α mentioned in Sec. 2.2 works only if all the characters in the document belong to one category. However, discussing such situation is nonsense. Thus, we discuss a method to distinguish characters into categories.

The most easily conceived method might be character recognition. However, recognizing distorted characters is not easy task. In addition, since just classification is required, recognizing (labelling) characters is unnecessary. Thus, we propose a novel classification method of characters. That is, classification by area ratios of character regions. The area ratio is known as an affine invariant. Though an affine invariant is not an invariant against perspective distortion, an affine invariant in a small region can be approximated as a projective invariant.

The area ratio has to satisfy the following two conditions. (1) The identical regions must be extracted.

Figure 4. An example of pairing of two characters. This is the case of "t" and "h."

Since an area ratio is invariant, an (approximately) same value is calculated at the same region. However, if the same region is not extracted due to perspective distortion, the invariant cannot be obtained. To avoid the problem, the proposed method introduces the area of a convex hull which can be calculated in the same manner under linear transformation. Thus, the ratio of two areas, foreground region of a character and its convex hull, is used.

(2) Area ratios must be distinctive enough to classify characters.
Characters of different categories whose area ratios are identical can exist. In this case, they must be misclassified into a cluster. In order to avoid bad influence, we use several area ratios at the same time. This increases the classification performance since the probability that several area ratios are simultaneously identical is less than the probability that one area ratio is identical. Since the number of areas calculated from one character is limited, we calculate area ratios by pairing two adjacent characters. Fig. 4 is an example of the pairing. At most five area ratios are calculated from every pair of adjacent two characters. Fig. 5 shows the five area ratios of two characters of Fig. 4. When m out of five area ratios are used for clustering, they are used as an m-dimensional feature vector. We employ k-means clustering algorithm to distinguish the vector. Each cluster is expected to contain each character pair in ideal. Hereafter, all the processes are performed for each pair of two characters.

Here, we mention the restriction of the proposed method on the size of characters. The relationship between the area and depth derived in Sec. 2.2 bases on the assumption that characters in the same category are the same size. Thus, if characters of different sizes exist in a document, the clustering using area ratios cannot distinguish the difference in size of characters. This can cause the estimation error of depth. However, the characters in different sizes can be eliminated by noise reduction process detailed in Sec. 2.4 since most documents consist of many body text characters in the same size and few characters in headings in different sizes.

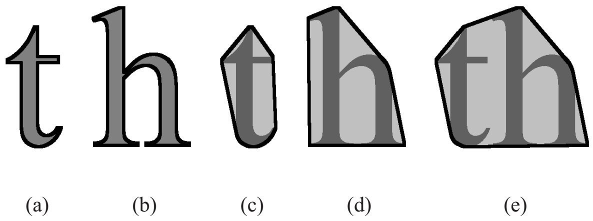


Figure 5. Five area ratios calculated from the pair of Fig. 4. (a) Area of "t," (b) Area of "h," (c) Area of convex hull of "t," (d) Area of convex hull of "h," and (e) Area of convex hull of "t" and "h."

2.4. Fitting to plane

The clustering procedure described in Sec. 2.3 enables us to fit character pairs of each category to each plane, respectively. The estimated plane can be different since the area of a character of the same size differs by category and estimated depth varies on the size. However, these plane must be the same one. Thus, in order to estimate the slant angle $alpha$ of the plane accurately, we integrate them. In Sec. 2.2, we let S be a known area of a character. However, it is actually unknown and differs by category. Thus, we estimate the area of a character S and the slant angle of only one plane $alpha$ simultaneously. In order to that, we replace S, $(X_j, Y_j, Z_j)^T$ and Z'_j appeared in Sec. 2.2 with S_i, $(X_{ij}, Y_{ij}, Z_{ij})^T$ and Z'_{ij} by adding cluster number i. We also add cluster number i to from Eq. (6) to Eq. (8).

To begin with, by substituting Eq. (6) into Eq. (7), we obtain

$$\begin{pmatrix} X_{ij} \\ Y_{ij} \\ Z'_{ij} \end{pmatrix} = \sqrt{S_i \cos\alpha} \begin{pmatrix} x_{ij}/\sqrt{s_{ij}} \\ y_{ij}/\sqrt{s_{ij}} \\ 1/\sqrt{s_{ij}} \end{pmatrix}. \tag{9}$$

This means that the coordinate $(X_{ij}, Y_{ij}, Z'_{ij})$ of each character is calculated by x_{ij}, y_{ij} and s_{ij} obtained from the image. However, since the inherent area of a character S_i in Eq. (9) and the slant angle α are unkonwn, we let $K_i = \sqrt{S_i \cos\alpha}$ and define an error of Z coordinates between the plane and a character of a category as

$$\begin{aligned} \varepsilon_{ij} &\equiv \left|\{aX_{ij} + bY_{ij} + c\} - Z'_{ij}\right| \\ &= \left|\left\{a\frac{K_i x_{ij}}{\sqrt{s_{ij}}} + b\frac{K_i y_{ij}}{\sqrt{s_{ij}}} + c\right\} - \frac{K_i}{\sqrt{s_{ij}}}\right|. \end{aligned} \tag{10}$$

Then, the sum of the error for all the characters of all the categories is given as

$$E = \sum_i \sum_j \varepsilon_{ij}. \tag{11}$$

Finally, we estimate the parameters $\{K_i\}$, and a, b and c which minimize Eq. (11). Note that since there is linear ambiguity in $\{K_i\}$, we fixed as $c = 1$, and estimate $\{K_i\}$, a and b in this paper. However, to estimate the plane, we cannot avoid taking the effect of noises (outliers) into account. The noises come from failure of extracting characters from the image, misclassification, and existence of characters in different sizes mentioned in Sec. 2.3. To deal with the noises, we perform two noise removal procedures: (A) removal of outliers, and (B) removal of clusters of outliers. The former removes a character where the error ε_{ij} is not less than a threshold t_1. The latter removes a cluster whose number of elements is not greater than t_2, since the cluster seems to estimate a wrong plane.

2.5. Rotation

We discuss how to set the view point in right front of the paper. This is performed by rotating the paper so that the view point moves on the normal of the paper. To represent a rotation matrix, we use the roll-pitch-yaw rotation angles where rotation angles around the X-, Y- and Z-axes are represented by ψ, θ and ϕ, respectively. The rotation matrix is given as

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{R}(Z,\phi)\boldsymbol{R}(Y,\theta)\boldsymbol{R}(X,\psi) \\ &= \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &\quad \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}. \end{aligned} \tag{12}$$

Since we defined $Z' = Z/f$, the equation of the plane $Z' = aX + bY + 1$ is $Z = afX + bfY + f$. The normal of the plane $Z = afX + bfY + f$ is $(af \;\; bf \;\; (-f))^T$. Thus, the rotation matrix $\boldsymbol{R}$ should be determined so that both the X and Y coordinates become 0. Since the rotation around the Z-axis is ignored, rotation angles of $\boldsymbol{R}$ are given as

$$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} = \begin{pmatrix} 0 \\ -\tan^{-1}\frac{af}{\sqrt{1+(bf)^2}} \\ -\tan^{-1}(bf) \end{pmatrix}. \tag{13}$$

Eq. (13) shows that an unknown parameter f is required for the estimation of the angles. This is in the same situation as [4] and [8]. Since f is not obtained in this paper, we simply let $f = 1$. Due to this, affine distortion remains after the rectification. The affine distortion is the distortion where parallel lines are kept parallel after transformation.

Finally, we rectify the image by the rotation. By rotating a point $(X_{ij}, Y_{ij}, Z_{ij})^T$ in the camera-centered coordinate system, we obtain the coordinate

$$\begin{pmatrix} \tilde{X}_{ij} \\ \tilde{Y}_{ij} \\ \tilde{Z}_{ij} \end{pmatrix} = \boldsymbol{R} \begin{pmatrix} X_{ij} \\ Y_{ij} \\ Z_{ij} \end{pmatrix}. \tag{14}$$

Then, projecting the point into an image plane, the coordinate on the 2D image plane

$$\begin{pmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{pmatrix} = \frac{f}{\tilde{Z}_{ij}} \begin{pmatrix} \tilde{X}_{ij} \\ \tilde{Y}_{ij} \end{pmatrix} \tag{15}$$

is given.

3. Experiment

We performed experiments to evaluate the proposed method. Five combinations of area ratios we used were the following:

i. $\dfrac{\text{Smaller (Fig. 5(a))}}{\text{Smaller CH (Fig. 5(c))} + \text{Larger CH (Fig. 5(d))}}$,

ii. $\dfrac{\text{Larger (Fig. 5(b))}}{\text{Smaller (Fig. 5(a))} + \text{Larger (Fig. 5(b))}}$,

iii. $\dfrac{\text{Smaller (Fig. 5(a))}}{\text{Larger CH (Fig. 5(d))}}$,

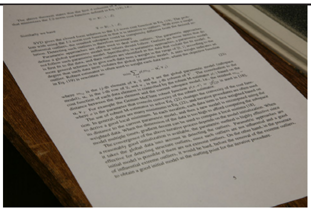
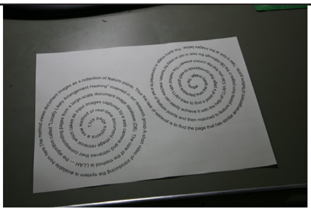
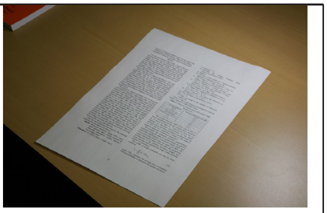
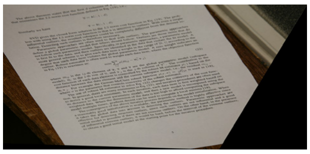
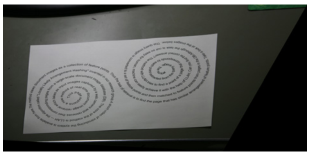
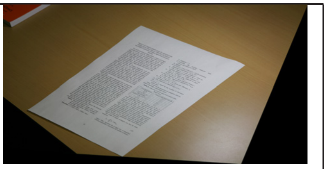
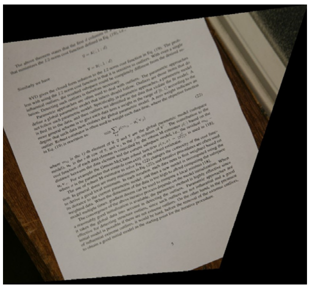
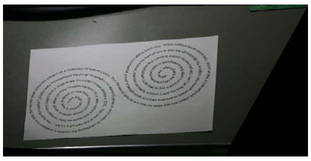
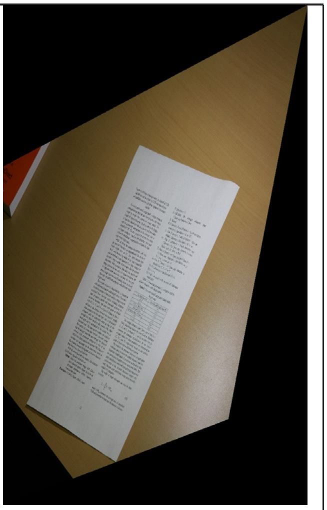
iv. $\dfrac{\text{Smaller CH (Fig. 5(b))}}{\text{Larger (Fig. 5(c))}}$,

v. $\dfrac{\text{Smaller (Fig. 5(a))}}{\text{Larger (Fig. 5(b))}}$,

where "smaller" stands for the area of smaller character of the pair, "larger" stands for the area of larger character, and "CH" stands for convex hull. When the dimensionality of an invariant vector was 5, area ratios i.∼v. were used. When the dimensionality of an invariant vector was 3, area ratios i.∼iii. were used.

The experimental results for three images are shown in Table 1. The images were taken by Canon EOS 5D, and their sizes were $4,368 \times 2,912$. Though the proposed method does not require a paper frame, for ease of visual evaluation, document images with paper frames were used. The images in the row (A) are original images. Those in the row (B) are rectified images with $f = 1$. Parameters for clustering and noise reduction were tuned so as to obtain the best result. In theory, the parallelism of two lines

Table 1. Rectification results and parameters of the proposed method. (A) Before rectification. (B) After rectification ($f = 1$). (C) After rectification (the best value of f is selected by hand).

		Image 1	Image 2	Image 3
(A)				
	Angles of two lines in long side (L) and narrow side (S)	(L) 8.98° / (N) 5.18°	(L) 1.73° / (N) 19.2°	(L) 6.64° / (N) 7.70°
(B)				
	Angles of two lines in long side (L) and narrow side (N)	(L) 0.06° / (N) 0.73°	(L) 0.14° / (N) 2.81°	(L) 1.78° / (N) 5.43°
	No. of dim. of invariant vector	3	3	5
	No. of clusters	60	40	200
	Thresholds t_1 and t_2	0.05, 20	(No thresholding)	0.1, 20
(C)		 $f = 11000$	 $f = 2000$	 $f = 15000$
	Angles of two lines in long side (L) and narrow side (N)	(L) 0.78° / (N) 0.02°	(L) 1.24° / (N) 1.09°	(L) 1.20° / (N) 3.88°
	Average difference of corner angles from right angle	0.605°	1.35°	4.16°

is rectified, though a right angle of a corner is not. In the experimental results, parallel lines of image 1 were rectified within 1°, and those of images 2 and 3 were not. The main cause is estimation errors of parameters of the plane due to outliers. The images in the row (C) are rectified images with the best f. In theory, not only the parallelism but also corner angles are rectified. In the experimental results, as the same reason as the row (B), both the parallelism and corner angles of image 1 was almost rectified, and that of images 2 and 3 were not. Therefore, improving estimation accuracy of a plane and deriving estimation method of f are required.

4. Comparison with existing methods

Since rectification of perspective distortion is a basic task on camera-based document analysis, there are many existing methods. They are roughly classified into the following three approaches: (1) using a paper frame, (2) using text lines, (3) using stereo vision. We mention the outline of the methods and discuss the difference from the proposed method.

The first approach assumes that the paper frame is a rectangle in nature and the frame can be clearly obtained. A rectangle suffering from perspective distortion becomes a quadrilateral since the parallelism of lines is lost. Using the information that the quadrilateral is originally a rectangle, we can calculate the transformation and rectify by the inverse transformation. This approach is used in [2] and some commercial products such as Ricoh Caplio R6. Though the approach is reasonable since many paper frames are rectangle, the whole document image have to be captured.

The second approach assumes the parallelism of text lines. For example, in [2], vanishing points are estimated from text lines, and then slant angles of the paper are estimated from the vanishing points[3]. This method first extracts text lines from a document image, and horizontal vanishing points are estimated. Next, assuming both ends of text lines are aligned vertically, vertical vanishing points are estimated by drawing three lines at right end, center and left end of the text lines. Finally, the document image is rectified by the two vanishing points and prior knowledge that horizontal text lines and vertical lines of ends are orthogonal[4]. The biggest drawback of the method is strong restriction for page layout. The method is not applicable to a document with complex layout and a document including many figures and equations since estimating both ends is difficult.

Approaches (1) and (2) mentioned above are not applicable to a document with complex layout, such that text lines are not parallel, and whole paper frame is not captured, such as the document image shown in Fig. 1.

The third approach estimates 3D shape of a paper with multiple cameras [5] or a movie capturing a document by a hand-held camera [9]. These require different devices from the proposed method.

5. Conclusions

In this paper, we proposed a method of rectifying perspective distortion into affine distortion without any prior knowledge. The proposed method estimates relative depth of each region of a document by employing an area of a character as a variant and an area ratio as an invariant. Then, 3D pose (slant angles) of the plane of the document is estimated. Since the proposed method does not use strong restriction on layout and way of capturing, it is applicable to document images of wide variety.

In the experiments, we confirmed the rectification ability of the proposed method in recovering the parallelism of lines. Though we confirmed that the rectification roughly succeeded, estimation accuracy should be improved. It can be achieved by employing the robust estimation and improving the performance of noise removing.

Due to the limitation of the proposed method, it cannot recover corner angles of a paper in principle. The cause is that there is linear ambiguity in depth estimation. This may be solved by employing other pair of variant and invariant. This is included in future work. Larger scale evaluation is also included in future work.

References

[1] *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[2] P. Clark and M. Mirmehdi. Recognising text in real scenes. *Int'l Journal of Document Analysis and Recognition*, 4:243–257, 2002.

[3] D. Doermann, J. Liang, and H. Li. Progress in camera-based document image analysis. pages 606–616, 2003.

[4] Y. ichi Ohta, K. Maenobu, and T. Sakai. Obtaining surface orientation from texels under perspective projection. In *Proc. of 7th International Conference on Artificial Intelligence*, pages 746–751, 1981.

[5] C. H. Lampert, T. Braun, A. Ulges, D. Keysers, and T. M. Breuel. Oblivious document capture and real-time retrieval. In *Proc. First Int'l. Workshop on Camera-Based Document Analysis and Recognition*, pages 79–86, Aug. 2005.

[6] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. *International Journal of Document Analysis and Recognition*, 7:84–104, 2005.

[7] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, June 2005.

[3] [2] includes two methods in approaches (1) and (2).

[4] Note that the prior knowledge used here, an original angle of horizontal and vertical lines, has the same amount of information as the focal length f because both of them reduce one degree of freedom of a projection matrix. If we use such a prior knowledge, we can rectify remaining affine distortion of the experimental result (B) and obtain (C) in Table 1.

[8] M. Pilu. Extraction of illusory linear clues in perspectively skewed documents. In *Proc. Computer Vision and Pattern Recognition, 2001 (CVPR '01)*, volume 1, pages 363–368, 2001.

[9] T. Sato, S. Ikeda, M. Kanbara, A. Iketani, N. Nakajima, N. Yokoya, and K. Yamada. High-resolution video mosaicing for documents and photos by estimating camera motion. In *Proc. SPIE Electronic Imaging*, volume 5299, pages 246–253, Jan. 2004.

Multimedia scenario extraction and content indexing for e-learning

Thomas Martin[1,2], Alain Boucher[3], Jean-Marc Ogier[1], Mathias Rossignol[2], Eric Castelli[2]

[1]L3i - Univ. of La Rochelle
17042 La Rochelle cedex 1
La Rochelle, France

[2]MICA Center
C10, Truong Dai Hoc Bach Khoa
1 Dai Co Viet
Hanoi, Vietnam

[3]IFI-MSI
ngo 42 Ta Quan Buu
Hanoi, Vietnam

thomas.martin@mica.edu.vn, alain.boucher@auf.org, jean-marc.ogier@univ-lr.fr
mathias.rossignol@mica.edu.vn, eric.castelli@mica.edu.vn

Abstract

In this paper, we present the use of multimodal content analysis in the MARVEL (Multimodal Analysis of Recorded Video for E-Learning) project. In this project, we record teachers giving their lectures in class and semi-automatically analyze the video-audio content in order to help transfer this lecture into a multimedia course for e-learning. We distinguish two primary goals in this application: scenario extraction (mostly from video) and content indexing (mostly from text and speech). Three objects take place in these goals: the teacher, the screen (for slide projection) and the whiteboard (for handwriting). These goals and the roles of all objects are explained in details, as well as our preliminary results. Through this application, we are giving some ideas about multimodality analysis and its formalization.

1 Introduction

Nowadays, as the available multimedia content grows every day, the need for automatic content analysis is becoming increasingly important. For example, information retrieval in broadcast news archives requires indexing the different medias available. Many projects currently focus on these research topics (content analysis, media enrichment...) but most of these works are focused on one single media, and are unaware of other medias. Because information is not concentrated in one media but distributed among all the medias, such approaches are losing important parts of that information, and ignore media interactions. Recently, many research works[14] have focused on the use of multiple modalities to increase the potentiality of analysis. However, to our knowledge, there is no existing framework for multimodal analysis, and there is only little serious study of the possibilities of interaction between modalities. In this paper, we present our ideas and framework on multimodal analysis, followed by our application in e-learning with the MARVEL project, which is divided into two goals: scenario extraction (mostly from the video) and content indexing (mostly from text and speech). This is still an on-going project, with some parts more developped than others. For each section, we will present our main ideas or detailed results, depending on work achievement.

2 multimodality

There is often confusion in the literature between the concept of media and the concept of modality. In many papers, the authors use both words to refer to the same concept. This does not seem to be exact, as we can see the two different concepts in the context of content analysis. We propose to define a modality as a refinement of the media concept. A media is characterized mostly by its nature (for example audio, video, text), while a modality is characterized by both its nature and the physical structure of the provided information (for example video text *vs* motion). One media can then be divided into multiple modalities, following two criteria: the semantic structure of the information and the algorithms involved in the analysis process. While the concept of media is independent from the application, the concept of modality is application dependant.

As proposed in [7], we will use generic modalities listed in three main families. First, the audio family includes different modalities in terms of structure such as speech, music or sound. Second, we distinguish between still image and motion (video) in the visual family. While both are acquired from a camera, motion contains time structure and is richer in term of content than still images. Third, the text family includes printed text and handwritten text.

This split of media into modalities can surely be discussed and different organizations can be proposed. We will use this scheme through this paper using several examples taken from some applications to illustrate our choice. We insist on the fact that the information contained in each modality has a different structure, regarding the algorithms that can be used, the difficulty for content extraction and for the semantic that can be given to it.

Once modality is defined, the next step is to define multimodality. In video indexing context, Snoek and Worring [14] have proposed to define multimodality from the author's point of view: it is "the capacity of an author of the video document to express a semantic idea, by combining a layout with a specific content, using at least two information channels". The inter-modal relation is then located at a high level using semantics. On the contrary, in the context of speech recognition, Zhi *et al.* [20] have implemented the multimodal integration just after the feature extraction phase and an alignment step. In this case, multimodal integration takes place at a low level. Both these definitions are incomplete. Furthermore, several multimodal applications found in the literature use two modalities, audio and video, and the multimodal part of these application is often limited to a fusion step. Examples of such works include applications for video indexing such as [17] where a high level fusion step is processed after speaker segmentation in audio and shot detection in video. Shao *et al.*[13] have performed multimodal summary of musical video using both audio and video contents. In the same domain, Zhu *et al.*[21] perform video text extraction and lyrics structure analysis in karaoke contents using multimodal approaches. Song *et al.*[15] recognize emotions using a fusion step just after feature extraction in audio and video. Zhu and Zhou [22] combine audio and video analysis for scene change detection. They have classified audio shots into semantic types and process shot detection in video They integrate then these results to have robust detection. Zhi *et al.*[20] and Murai *et al.*[10] use facial analysis (video) to improve speech recognition (audio). [10] detects shots in video containing speech whereas [20] combines lip movements and audio features to process speech recognition. Zotkin *et al.*[23] propose a tracking method based on multiple cameras and a microphone array. Bigün *et al.*[4] proposed a scheme for multimodal biometric authentication using three modalities: fingerprint, face and speech. Fusion is processed after individual modality recognition.

We propose a more general definition of multimodality as an interaction process between two or more modalities. This process is based on an inter-modal relation. We have identified three different types of inter-modal relations [8]: trigger, integration and collaboration. The trigger relation is the simplest relation: an event detected in one modality activates an analysis process to start in another modality. The integration relation is already widely used and is mainly characterized by its interaction level. The analysis processes are done separately for each modality, but followed by a process of integration (fusion or others) of their results. Snoek and Worring [14] present a more complete review of existing works widely using the integration relation for the application of multimodal video indexing. The third relation is collaboration, and it is the strongest multimodal relation, consisting in a close interaction of two modalities during the analysis process itself. The results of the analysis of one modality are used for analyzing a second one.

3 Video analysis for e-learning

Our main application for multimodality is e-learning through the MARVEL project. The goal of MARVEL (Multimodal Analysis of Recorded Video for E-Learning) is the production of tools and techniques for the creation of multimedia documents for e-learning.

The complete course of a professor is recorded live. Furthermore, textual sources such as course slides may be available. The recorded material from live courses is analyzed and used to produce interactive e-courses. This can be seen as an application of video analysis to produce rich media content. The slides used by the professor in the class can be automatically replaced by an appropriate file in the e-course, being synchronized with the professor's explanations. The course given by the professor is indexed using various markers from speech, text or image analysis. The main aim of this project consists in providing semi-automatic tools to produce e-learning courses from recorded live normal courses.

In this project, three different medias are available: audio, video and lecture material (essentially the slides). Following the model proposed in section 2, we have identified five different modalities: *i) printed text* which contains text from the slides and, if available, from other external textual sources. This modality is present in both video and lecture material media; *ii) handwritten text* which consists in the text written on the whiteboard; *iii) graphics* which include all the graphics and images present in the slides. *iv) motion* which contains the motion content of the video media; *v) speech* which gathers the teacher's explanations.

To simplify the explanations in this paper, we will not take into account the *graphic* modality and we consider only the textual parts of the slides. A difference must be made between *handwritten text* and *printed text* for two reasons. First, as presented in section 2, the nature of both modalities is different (handwritten *vs* printed text). The second reason is specific to this application: the two modalities do not contain the same data. Even if the contents of both modalities are related to the course, one (*printed text*) is more structured than the other.

The *printed text* modality is available in two different medias: video and text. It is a good example to illustrate our distinction between media and modality (section 2). Even if it is available in two different medias, the *printed text* still contains the same information, with the same structure. Once detected and extracted from the video media, the analysis processes involved are similar whatever the media.

The application is divided into two distinct parts, which represents two different, but complementary, goals to achieve: *i scenario extraction* (section 4): The scenario is given mainly by the video. The teacher's behavior (see *fig.* 1) is analyzed to extract the course scenario (explaining the current slide, writing on whiteboard, talking to the class,. . .). This will be used later as a layout during the e-course production. Other regions of interest such as the screen or the whiteboard are detected. Detections of slide changes or new writing on the whiteboard are events that will be used; *ii content indexing* (section 5): The content indexing of available media has to be done using the speech given by the teacher, the printed text on the slides and the handwritten text on the whiteboard. These three sources are complementary to show all the content of the course. Different inter-modal interactions are identified here.

During the first part of the application (scenario extraction), 3 trigger relations are involved. These relations are directly related to the actors who interact in a course: teacher, whiteboard and screen. The trigger source is the *motion* modality. First, the "slide transition" event triggers the *printed text* detection and recognition. Second, the "teacher points at screen" event triggers the point of interest search. Third, similar to the first, the "teacher writes on whiteboard" event triggers the *handwritten text* recognition process.

The second part of the application (content indexing) contains most of the inter-modal relations. First, the *speech-printed text* interaction is a bimodal and bidirectional collaboration interaction, with its main direction from *printed text* to *speech*. As used in [20, 10], *motion-speech* interaction can be also useful . Recognition of *handwritten text* is a difficult task, especially in video. We propose to help recognition of *handwritten text* using both *speech* and *printed text* modalities. Both relations, *speech-handwritten text* and *speech-printed text*, are bimodal and unidirectional.

4 Scenario extraction

Scenario extraction aims at retrieving the structure of the lecture. We have identified three elements in the MARVEL application (see *fig.* 1: the screen, the whiteboard and the teacher. Both the screen and the whiteboard are passive elements, whereas the teacher interacts with the others. The screen is the support to display slides. In the rest of this paper both words, screen and slides, will be used referring to the same object. The information displayed on the screen is structured and contains text and graphics.

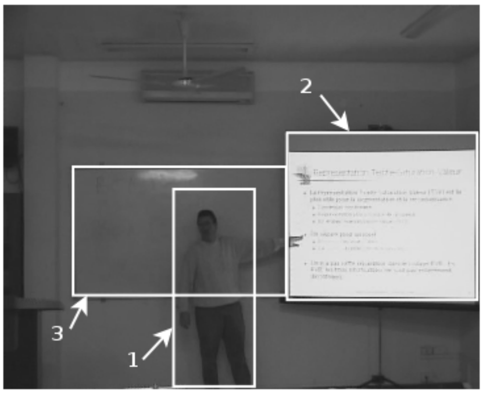


Figure 1. Frame extracted from a recorded course. White shapes highlight identified actors of the application: the teacher (1), the screen (2) and the whiteboard (3).

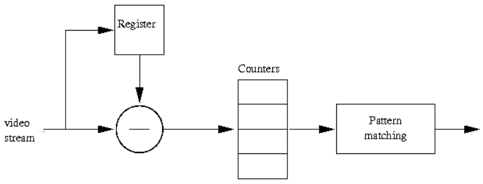


Figure 2. Slide change detection schema

4.1 Slide change detection

Similarly to shot detection in more general videos, slide change detection aims at segmenting the video into logical units, each containing a different slide. A slide change is defined as the replacement of a slide by another. Such change is characterized by an image modification in the screen zone. This modification is more or less important according to the considered slide change, and can be considered as the result of an edit operation on the current slide. Slide changes have a global effect to the screen, whereas slide modifications are more located.

During the course the teacher's interactions with the screen can temporally occlude the slide. Another source of motion is inherent to the compressed video: as video compression algorithms often suppress high frequency information, small patterns such as letters are affected by temporal noise. Such patterns are obviously frequent in slides.

Our slide change detection algorithm is based on image differences. However, we introduce a priori knowledge

sequence	occured	detected	false detections
seq1	24	23 (98.8%)	0
seq2	10	10 (100%)	1
seq3	13	13 (100%)	0

Figure 3. Some results for our slide transition detection algorithm on three video sequences taken from three different class courses. For each sequence, the actual number of transitions occurring in the sequence is shown, followed by the numbers of detected transitions and false detections.

by restricting this to the slide zone in the stream. As the camera position and view are fixed during the recording, we manually fix this zone. That screen zone is extracted from each 25th frame (number decided upon experimentally), and an image difference is computed. The resulting image is thresholded to eliminate the compression noise. To avoid problems due to the teacher's interaction with the slide, we divide the image into quarters and count the modified pixels in each quarter. Indeed, modifications will be detected when the teacher interacts with the screen. However, if the screen is divided in 4 parts, he will not interact with all quarters at the same time. The 4 resulting values are normalized. Thus, we obtain 4 temporal curves describing motion in the slide zone.

The slide transition detection is performed through simple pattern matching on these curves. If modifications are simultaneously detected in the 4 quarters within less than 2 seconds, we consider that a slide transition has occurred. Simultaneous modifications in one, two or three quarters are pieces of evidence but are not sufficient to detect slide modification.

Tests have been performed on three sequences (see table 3). These results are quite satisfactory. However, in the case of slide changes occurring on three or less quarters of the slide, the algorithm will not detect them. We will see in the next section that the teacher detection algorithm permits to solve this problem.

4.2 Teacher detection

The teacher detection aims at getting the position of the teacher in the classroom and to determine with which zone he is interacting: the screen or the whiteboard. For this task, we use an algorithm based on image difference (see *fig.* 4). To improve its results, we extract an image of the background, which is substracted from the current frame. However, the screen zone in the image difference is very noisy and causes many false detections. To avoid this, we

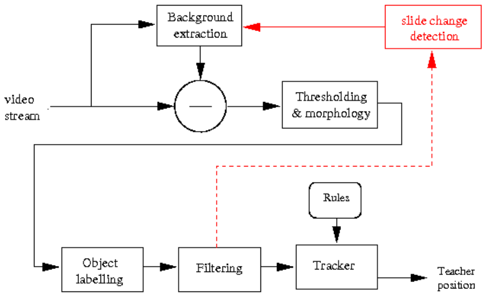


Figure 4. Teacher detection schema

use the slide transition detection results (see section 4.1) to correctly update the background. More precisely, the screen zone of the background is not updated, except if a slide change is detected. After a morphological step, bounding-box candidates are extracted and filtered. Collaborations between these two algorithms lead to more accurate results for the course scenario extraction.

4.3 Teacher gesture detection

Depending on the people, gesture may take an important part in the communication process. In our specific context, two main groups of gestures can be identified: *i)* free gestures; as an example when the teacher is interacting with the classroom, *ii)* constrained gestures; we put in this group the interactions with the whiteboard or the screen.

Even if gestures can provide useful information, due to high variability of gesture types, the first category does not seem to be usable. On the contrary, the second one does not have this problem, and specific gestures, such as pointing something, can be identified. Moreover, the succession of gestures can provide relevant indices for whiteboard or slide content ordering (scenario extraction giving the order used by the teacher to present the content).

5 Content indexing

5.1 Text detection and recognition

As presented, our strategy relies on the inter-modality cooperation for the course content indexing process. Here, the purpose is to use text recognition from the slides in the video for guiding the indexing process, especially through providing information from this text recognition module to the speech recognition one.

The problem of text recognition is a very well known problem, for which many contributions can be found in the litterature [6], and industrial software quite reliable. Most of these recognition tools have been developed in the context of "high resolution" images, and have to be re-visited and adapted in the context of our problem, because of the quality of the images.

In the context of the MARVEL project, two categories of text information have to be considered: the text which is handwritten by the teacher on the blackboard and the text which is presented on slides prepared on ICT tools such as PowerPoint. The indexing process can also rely on graphic parts, drawn by the teacher on the blackboard or presented on the slides. So far, we have not considered the question of the recognition and indexing of all the information drawn or written by the teacher on the blackboard.

Our first developments deal with the slides-based indexing process, through a recognition process of the information which is presented on these supports. In this kind of context, the usual document processing chain proposes a first stage whose aim is to separate (segmentation stage) the different layers of information of the document. Generally this "physical" segmentation process depends on the a priori knowledge concerning the information of the document: text, size... In the context of our project, we have decided to apply a blind segmentation process, based on very relevant tools developped in the context of ancient document processing [5]. The segmentation process relies on the computation of the auto-correlation function, allowing to detect regular orientations, some of them being highly representative of the presence of text.

Using these tools in the context of slide segmentation is found to be a very relevant approach, since it is very difficult to have reliable a priori knowledge concerning text features. Concerning the text recognition engine, we decided to develop our own recognition tools. This decision was motivated by a strong competence in this domain in our lab, and also because we wanted to take the benefits of all the intermediate information concerning the recognition process, which is rarely available in industrial tools. As a consequence, we developed a "classic text recognitionengine", based on relevant features [11]. These features are introduced as input of a KNN classifier [1], allowing to provide a confidence associated with each decision, information that can be re-used in a feedback process, in the context of inter-modality cooperation. The exploitation of this text recognition tool in the context of slide recognition is very encouraging.

A syntactic analysis tool allows to increase this recognition rate, in relation with a dictionary which is available in our system. This text recognition tool provides some information that can be considered as indexes for the indexing process, and that can be transmitted to the speech recognition module to increase its performances. This intermodality indexing process allows to increase the quality of the index in a very significant manner.

5.2 Speech recognition

In the MARVEL project, we aim at indexing available data streams for further use such as audio-video and slides synchronization. The most direct way to obtain semantic indexing is through linguistic data, which can in particular be obtained using speech recognition techniques. However, in such a project, full continuous speech recognition is not useful, since we do not intend to perform a complete automatic transcription, but only audio content indexing. Thus, our aim is to detect keywords in the speech recording. In a first research step, we perform tests in order to evaluate what we can recognize using an existing automatic speech recognition (ASR) tool. The ASR software used is Raphael [2], which is *a priori* not well adapted at all to the kind of speech we are dealing with, but is quite representative of the state of the art in voice analysis.

ASR tools typically use a three step process. First, potential phonemes are extracted from the signal using an acoustic model, then a lexicon of phonetic transcriptions is consulted to find which words may correspond to those phonemes, and finally the lattice of word hypotheses is processed through a language model (LM) to find which sequence of words is the most linguistically plausible one. We cannot affect directly the first of those steps, since developing acoustic models is a huge, very technical task, but the two subsequent ones exhibit weaknesses which we can amend. A first problem is the incompleteness of the phonetic lexicon, from which words used during the course may be absent. Since the absent words are typically the most specialized, technical ones, which are also the most likely to be interesting keywords, this is a very critical problem. A second difficulty appears with the language model: in the tool we are using, as in most existing ASR software, the LM consists in a database of three-word sequences (3-grams) probabilities. Such probabilities are difficult to compute reliably for general spoken language—if such a thing even exists—and in the ideal case, a specialized language model adapted to the considered speaker and topic must be used. We must find an inexpensive way to develop such a specialized LM without the data usually exploited or that task (a consequent transcript of spoken language dealing with the considered topic).

The chosen approach is to mix an existing generic spoken language model with a small specialized language model extracted from textual data related to the course: the text of the slides used by the teacher. This text is very simple and features prominently the keywords of the lessons, which is precisely what we are interested in. From the same text, we shall also extract all words that are absent from the phonetic lexicon and add them to it thanks to an automatic phonetization tool.

A similar idea is followed in [18], but the authors of that

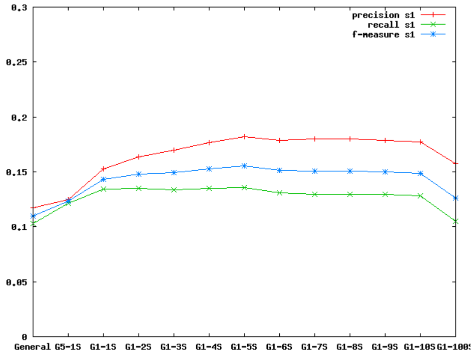


Figure 5. Speech recognition results for all words with different models mixing weights being used. In that case, recognition rates vary between 10% and 20%.

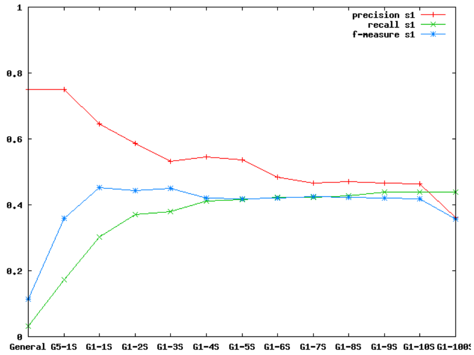


Figure 6. Speech recognition results restricted to keywords extracted from slides with different model mixing weights being used. On these domain keywords, results are significantly higher and vary around 50%.

work evaluate the interest of the additional textual information in terms of theoretic modeling power (perplexity) of the hybrid LM. We shall adopt a more practical approach, and directly evaluate the impact of that transformation of the LM on speech recognition rates. In order to perform our tests, we have used two sequences named *speaker1* and *speaker2*, each with a distinct speaker and topic. *Speaker1* is a 20 minutes presentation of approx. 3,500 words, whose attached slides gather 748 words, while *speaker2* lasts 35 minutes, counts 6,000 words, and has slides gathering 751 words.

The generic LM is a French spoken language model with a vocabulary containing about 16,500 words. The specialized LM is automatically built from the text slides using the SRILM language modeling toolkit [16], and the phonetization of new words is performed by the LIA-PHON automatic phonetizer for French [3]. Tests have been performed with different language models: *i* the original, generic model of Raphael only; *ii* various mixed models obtained by performing a weighted average of transition probabilities between the generic and specialized model with the following weights: 5-1, 1-1, 1-2…1-10, and 1-100 (first number is the generic model weight and second is the specialized model weight).

5.2.1 Results

The two sequences have been manually transcribed. For each sequence the recognition result has been aligned to this transcription. Recognition rates have been computed on: *i)* all words, *ii)* keywords manually selected by the teacher, representative of the course content.

Figures 5 and 6 present the results obtained for the sequence *speaker1* depending on the model mixing weights used. The results obtained for *speaker2* are nearly identical. We can observe that the use of a mixed model significantly improves the results of speech recognition for all words relatively to the original performance, but since that one was very low, the results remain not reliable enough to be exploitable. However, recognition rate on keywords is greatly improved. The best obtained f-measure using a mixed model is about 50% (with 1-4 and 1-5 weights) while it was only about 10% using the general model. In both cases (and on *speaker2*), we can see a peak in performance when the weight for the specialized LM is about 5 times higher as that of the general LM. That seems to correspond to an optimal level of specialization, above which the LM loses too much generality to be able to model “ordinary” speech.

5.3 A video-text driven speech keyword detection

As the results presented above show, the result of continuous speech recognition is not usable as is. However, the recognition results can be improved by introducing a specialized knowledge. In the context of the MARVEL project, such knowledge can be provided as automatically as possible. We propose to automatically select keywords in the slide text and to use them to improve speech recognition. Provided that the slides are not available as input of our process, their text has to be extracted from the video stream.

Instead of building a LM with this text, we propose to stop the continuous speech recognition process after phoneme extraction. At this step, the output is a lattice of phonemes hypothesis. Selected keywords will be searched in this lattice.

Keywords are selected in the text of the slides. Depending on the teacher, this text is more or less concise. The automatic keyword selection can be performed using a *stoplist* or more complex methods using morpho-syntactic analysis with tools such as *TreeTagger*[12]. After this selection step, these keywords will be phonetized using the LIA-PHON phonetizer, then searched in the phoneme lattice.

5.4 Text and speech attributes

Slide text information is not only borne by words. Text can have many attributes that participate in characterizing the content, such as size, position of text, style (title, subtitle or item), color, font weight, slanting, underlining, etc. These attributes can be used to order the different ideas presented in the slide and to stress on some important ideas. A complete slide text representation model must include these attributes. They will be used later for multimedia representation of the course content, but also for content retrieval (section 5.5).

Similarly to text, speech can also have many attributes to characterize its content. These attributes can be relative to the prosody of the speech or to emotions expressed when speaking. In the first case, changes in prosody can be used to determine between interrogative and affirmative sentences for example [9]. In the second case, emotions in the speech can be used to emphasize on a word or to discriminate between two ideas [19].

Text and speech attributes do not contribute to the content recognition process. They are mostly recognized independently and associated to the (spoken or written) words that they characterize, but they will be of importance as driving factors for the indexing and retrieval of the course contents.

5.5 Content indexing and retrieval

So far we have worked on speech and text detection and recognition, with some experiments on attribute recognition. The final objective of the content recognition for this application is to be able to index and retrieve the course content. A user (student for example) should be able to query a course database and retrieve links to audio-video records fulfilling his needs. In this section, we present our preliminary ideas on course content indexing and retrieval.

An accurate model for content indexing and retrieval must include four aspects: text, speech and their respective attributes. The speech recognition model based on text slides (section 5.2) is limited on domain key words. Following this model, content indexing is limited to key words, both for text and for speech, and do not include the whole word content. This limitation restricts the retrieval scheme to key words or key concepts in the course domain. It is acceptable for the application, where content indexing and retrieval should help the user to browse into the course content.

The currently developed content retrieval model is based on the combination of text and speech, plus attributes when available. The time unit used to index the text and speech content is based on slide change detection (section 4.1), which defines a time interval [t1,t2] for each slide. Text is naturally associated with slide display. But speech can also be indexed using the same scheme, given the hypothesis that the teacher's speech is always related to the displayed content. This hypothesis is not always true, but sufficiently to allow indexing of all speech content following that scheme. To be more specific, speech associated to a slide lasts from the last audio silence before the slide change (marked as the beginning of a sentence) to the first silence following the next slide change (marked as the end of a sentence).

Undergoing work bears on the weights in the retrieval model to be associated with text and speech attributes. It sounds natural that the co-presence of a word in both the speech and the text content indicates a high relevance of this part of the video regarding to the query. But the influence of attributes on the relevance of a word is less obvious, and depends on each attribute.

Regarding the text attributes, the position and the size of a word gives a good idea of its relevance. But other attributes such as color, bold or italic need to be tested, as there are no given rules on how to use these attributes. They also depend on each person who can mean different things using the same attributes. While some people use many attributes on their slides, others may never use them. Moreover, the attributes may not be on the indexed keywords, but on neighbor words. A possible example of this is the emphasis that can be made on some words like *do*, *must* or *never*, which are not domain keywords but used to characterize the text preceding of following them.

Regarding the speech attributes, emphasis on words seems to be the most important attribute to take into account. Such emphasized words should be more relevant. The role of other attributes in the retrieval model is not clear and not defined yet. As for the text attributes, the speech retrieval model should take into account emphasis made on generic words not part of the content domain but used to emphasis preceding or following domain content.

6 Conclusion and future works

In this article, we have presented our work on multimodal analysis through the two main goals of the MARVEL project: scenario extraction and content indexing. The interactions between the different modalities for the indexing process rely on a device based on different triggers allowing starting the cooperation between the different recognition modules. Of course, our future works will deal with the improvement of each recognition module, but the theoretical works will also consider a formal description of these interactions through adapted mathematical tools. For these points, some current studies deals with Petri nets combined with Bayesian networks.

So far, we have skipped the whiteboard text analysis and indexing. As it is handwritten text and low resolution images, this work is more difficult than for printed text.

As presented, we are currently developing a keyword detection tool. Our aim is to automatically select keywords from the slides. If an electronic version of the slides is available, direct access to the text is available. If not, we have to perform video text recognition to access this text. One possibility to select the keywords is to analyze de teacher's gestures. For example, when the teacher points at a zone in the current slide, that information can be used to characterize and to stress on a given content. Consequently, specific words can be highlighted and selected as relevant keywords.

References

[1] K. Aas and L. Eikvil. Text categorisation: A survey. Technical Report 941, Norwegian Computing Center, 1999.

[2] M. Akbar and J. Caelen. Parole et traduction automatique: le module de reconnaissance RAPHAEL. In *17th International Conference on Computational Linguistics (COLING 98)*, pages 36–40, Montreal, Québec, Canada, 1998.

[3] F. Béchet. LIA-PHON : Un système complet de phonétisation de textes. *TAL (Traitement Automatique de Langues)*, 42(1):47–67, 2001.

[4] J. Bigün, J. Fiérrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Multimodal biometric authentication using quality signals in mobile communications. In *Proc. of ICIAP*, pages 2–11. IEEE Computer Society, 2003.

[5] N. Journet, R. Mullot, V. Eglin, and J. Ramel. Dedicated texture based tools for characterisation of old books. *dial*, 0:60–69, 2006.

[6] L. Li, G. Nagy, A. Samal, S. C. Seth, and Y. Xu. Integrated text and line-art extraction from a topographic map. *IJDAR*, 2(4):177–185, 2000.

[7] T. Martin, A. Boucher, and J.-M. Ogier. Multimodal analysis of recorded video for e-learning. In *Proc. of the 13th ACM Multimedia Conference*, pages 1043–1044. ACM Press, 2005.

[8] T. Martin, A. Boucher, and J.-M. Ogier. Multimodal interactions for multimedia content analysis. In *Proc. of ICTACS 2006*, pages 68–73. World Scientific, 2006.

[9] V. Minh-Quang, T. Do-Dat, and E. Castelli. Prosody of interrogative and affirmative sentences in vietnamese language: Analysis and perceptive results. In *Interspeech 2006*, Pitsburg, Pennsylvania, US, 2006.

[10] K. Murai, K. Kumatani, and S. Nakamura. Speech detection by facial image for multimodal speech recognition. In *Proc. of ICME*, page 149. IEEE Computer Society, 2001.

[11] S. Nicolas, T. Paquet, and L. Heutte. Markov random field models to extract the layout of complex handwritten documents. In *IWFHR-10*, pages 563–568, La Baule, France, 2006.

[12] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[13] X. Shao, C. Xu, and M. S. Kankanhalli. Automatically generating summaries for musical video. In *Proc. of ICIP*, volume 2, pages 547–550, 2003.

[14] C. G. M. Snoek and M. Worring. Multimodal video indexing: a review of the state-of-the-art. *Multimedia Tools and Applications*, 25(1):5–35, 2005.

[15] M. Song, J. Bu, C. Chen, and N. Li. Audio-visual based emotion recognition - a new approach. In *Proc. of CVPR*, volume 2, pages 1020–1025. IEEE Computer Society, 2004.

[16] A. Stolcke. SRILM – an Extensible Modeling Toolkit. In *ICSLP 02*, pages 901–904, Denver, CO, USA, 2002.

[17] S. Tsekeridou and I. Pitas. Audio-visual content analysis for content-based video indexing. In *Proc. of ICMCS*, volume 1, pages 667–672. IEEE Computer Society, 1999.

[18] L. Villaseor-Pineda, M. M. y Gmez, M. Prez-Coutio, and D. Vaufreydaz. A Corpus Balancing Method for Language Model Construction. In *4th International Conference of Computational Linguistics and Intelligent Text Processing*, pages 393–401, Mexico City, Mexico, 2003.

[19] L. Xuan-Hung, G. Quenot, and E. Castelli. Speaker-dependent emotion recognition for audio document indexing. In *The 2004 International Conference on Electronics, Informations and Communications (ICEIC 2004)*, Hanoi, Vietnam, 2004.

[20] Q. Zhi, M. Kaynak, K. Sengupta, A. D. Cheok, and C. C. Ko. HMM modeling for audio-visual speech recognition. In *Proc. of ICME*, pages 201–204. IEEE Computer Society, 2001.

[21] Y. Zhu, K. Chen, and Q. Sun. Multimodal content-based structure analysis of karaoke music. In *Proc. of the 13th ACM Multimedia Conference*, pages 638–647. ACM Press, 2005.

[22] Y. Zhu and D. Zhou. Scene change detection based on audio and video content analysis. In *Proc. of ICCIMA*, page 229, 2003.

[23] D. Zotkin, R. Duraiswami, and L. S. Davis. Multimodal 3-d tracking and event detection via the particle filter. In *Proc. of Event*, 2001.

Demos

Automatic Identity Card and Passport Reader System

Vicente Chapaprieta, Josep Lladós, Felipe Lumbreras, Jordi López
Icar Vision Systems, S.L.
http://www.icarvision.com

Abstract

ICAR, Identity Card Automatic Reader, consists of a system for reading identity cards like passports, identity documents and driver licenses. First, document image is acquired by an own acquisition device based on a digital camera chip. Then, the image is analyzed in order to identify document type among a set of predefined models using local features analysis and text localization. Identification process is performed independently from image background. For unknown input documents, lines compliant with machine readable ICAO 9303 format are located and recognized. Once the document has been identified, textual fields are automatically located and read using an OCR based on statistical analysis. The system is currently installed as a check-in application in real environments like hotels and corporate buildings.

Demo Abstract: Semantic Annotation of paper-based Information

Heiko Maus, Andreas Dengel
German Research Center for AI
Knowledge Management Department
Kaiserslautern, Germany
{firstname.lastname}@dfki.de
http://www.dfki.de/km/

Abstract

The demo shows how a user is easily able to semantically annotate and enrich scanned documents with concepts from his personal knowledge space. This bridges the gap between paper documents and the approach of a Semantic Desktop, where each information object on the user's computer desktop is semantically described by means of ontologies.

1. Motivation

The demo of the SCETagTool shows how a user is easily able to semantically annotate and enrich scanned documents with concepts from his personal knowledge space. This bridges the gap between paper documents and the approach of a Semantic Desktop, where each information object on the user's computer desktop is semantically described by means of ontologies.

Thus, a user is able to connect with minimum effort the content of paper documents to concepts or topics he is dealing with because the SCETagTool proposes those concepts for text passages or words from the paper document. The user is then able to interact directly on the document image and accept those proposals.

The resulting document text with its connections to concepts is inserted as a Wiki page in the user's information system representing his personal knowledge space – the gnowsis Semantic Desktop [5]. Thus, the formerly passive paper document is now electronically available and embedded in the user's personal knowledge space, ready for later retrieval when searching, e.g., for documents dealing with specific concepts.

The presented system is realized as a service extending the gnowsis Semantic Desktop for introducing paper-based information objects into the personal knowledge space.

2. Flow of Work

The flow of work is as follows (see Fig. 1): The user scans a document, e.g., a newspaper article, with a document camera, the document image is automatically handed over to the SCETagTool[1], and OCR is applied to the document image.

The document image is shown in the SCETagTool. Now the user is able to select a title for the document. For each text passage selected by the user:

- the proposed concepts are shown directly on the document image by highlighting the words and listing the concepts beneath the mouse cursor, so the user can easily accept them if appropriate[2],
- get and accept tag proposals for the text passage,
- add existing or create new concepts the text passage.

If finished, the system creates a Wiki page[3] with the text of the scanned document and adds it to the gnowsis, including

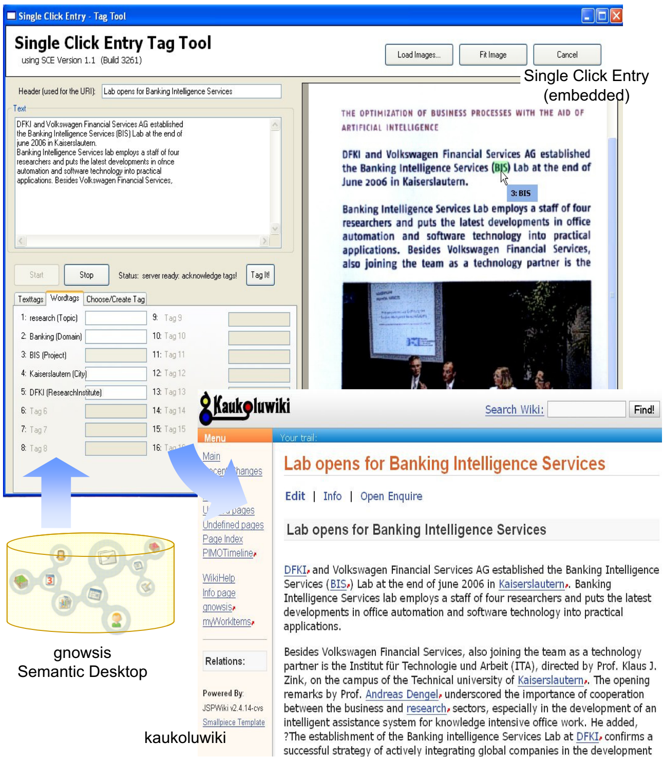
- embedded hyperlinks from the words to the concepts accepted by the user,
- concepts as tags for text passages (also with hyperlinks to the concepts),
- a hyperlink to the document image (which was saved in the user's file system).

The concepts proposed are taken from the user's PIMO (*Personal Information Model Ontology*) which consist of classes such as Person, Location, Document(-type), or Topics, and their respective instances – e.g., "Andreas Dengel"

[1] done by observing a file folder

[2] Functionality provided by Single Click Entry, see below.

[3] More precisely: it creates an instance of a the class Document and adds the text as content which is then displayed as a Wiki page.



Header (used for the URI): Lab opens for Banking Intelligence Services

Text: DFKI and Volkswagen Financial Services AG established the Banking Intelligence Services (BIS) Lab at the end of june 2006 in Kaiserslautern.
Banking Intelligence Services lab employs a staff of four researchers and puts the latest developments in ofnce automation and software technology into practical applications. Besides Volkswagen Financial Services,

Status: server ready: acknowledge tags!

1: research (Topic)		9: Tag 9	
2: Banking (Domain)		10: Tag 10	
3: BIS (Project)		11: Tag 11	
4: Kaiserslautern (City)		12: Tag 12	
5: DFKI (ResearchInstitute)		13: Tag 13	
6: Tag 6		14: Tag 14	
7: Tag 7		15: Tag 15	
8: Tag 8		16: Tag 16	

THE OPTIMIZATION OF BUSINESS PROCESSES WITH THE AID OF ARTIFICIAL INTELLIGENCE

DFKI and Volkswagen Financial Services AG established the Banking Intelligence Services (BIS) Lab at the end of June 2006 in Kaiserslautern.

3: BIS

Banking Intelligence Services Lab employs a staff of four researchers and puts the latest developments in office automation and software technology into practical applications. Besides Volkswagen Financial Services, also joining the team as a technology partner is the

Lab opens for Banking Intelligence Services

Lab opens for Banking Intelligence Services

DFKI and Volkswagen Financial Services AG established the Banking Intelligence Services (BIS) Lab at the end of june 2006 in Kaiserslautern. Banking Intelligence Services lab employs a staff of four researchers and puts the latest developments in office automation and software technology into practical applications.

Besides Volkswagen Financial Services, also joining the team as a technology partner is the Institut für Technologie und Arbeit (ITA), directed by Prof. Klaus J. Zink, on the campus of the Technical university of Kaiserslautern. The opening remarks by Prof. Andreas Dengel underscored the importance of cooperation between the business and research sectors, especially in the development of an intelligent assistance system for knowledge intensive office work. He added, ?The establishment of the Banking intelligence Services Lab at DFKI confirms a successful strategy of actively integrating global companies in the development

Figure 1. System overview of the SCETagTool

`isa Person` – found on the user's desktop. For instance, all users from the email address book (such as Outlook or Thunderbird) are instantiated as Persons. With this approach a user's personal knowledge space is represented in his PIMO which in turn serves as the base ontology for the gnowsis Semantic Desktop.

The resulting Wiki page is thus connected to the user's concepts such as persons, locations, companies, and topics. If the user browses, e.g., a person in the gnowsis, all connected Wiki pages are listed, i.e., also the page from the previously scanned paper document.

As the gnowsis is a Semantic Desktop, it is then possible to exploit those documents also by means of semantic search, e.g., provide all documents where persons from the company DFKI are mentioned.

3. System overview

The SCETagTool is a Java application consisting of several components which are commercial as well as open source:

The document image is delivered by the portable, digital document camera *sceye*[4] from the company *silvercreations*. With its pivot arm it can be quickly placed on a desktop and the document camera is ready to scan within seconds, thus it supports also mobile workers.

The OCR and the document image interaction is provided by the tool *Single Click Entry*[5] from *ODT* (*Océ Document Technologies*) which is a system for guided data capture from document images. Its function is as follows: Assume an insurance form which requires the input from an insurance claim letter. The required input of the specific fields are described by data format and regular expression (e.g., "US phone number"). Now, Single Click Entry guides the user through the displayed image and highlights the data found (e.g., a telephone number) for the current field (e.g., "phone number of policy holder") as well as lists the data in an information box at the mouse cursor. In case the data is correct the user accepts the proposal by simply clicking on it. In that case the data is transferred to the input data field and the data for the next field is highlighted. Single Click Entry provides several SDK's (e.g., VBS, C#) for including this into own applications what has been done for the SCETagTool.

The gnowsis Semantic Desktop is an open source[6] system which provides a semantic layer to information objects on the user's computer desktop. Embedded in the gnowsis is the kaukoluwiki – a Semantic Wiki which enables semantically enriched Wiki pages [3] and is based on the PIMO.

4. Outlook

This first prototype serves as a basis for further investigation of the easy document capturing interaction method provided by Single Click Entry and how to embed this more tightly into the gnowsis as a means for evolving the user's personal knowledge space with paper documents in order to evolve the approach presented in [4].

Furthermore, more work has to be spent on proposing adequate concepts for document text. Here, we will use the personalized, multi-perspective document classification approach as explained in [1] where the documents belonging to a concept are used to learn a document similarity vector for that specific concept. This is used in order to propose the concept for suitable text passages, i.e., if the passage is similar to the learned vector. Furthermore, more sophisticated proposals will be applied with a collection of specialized services which analyse the text and propose concepts from the PIMO [2].

Acknowledgements The work has been funded by the Rhineland-Palatinate cluster of excellence "Depend-

[4] `http://www.sceye.biz`

[5] `http://www.odt-oce.com/SCE/`

[6] Download at `http://www.gnowsis.org`

able adaptive systems and mathematical modeling" (DASMOD), project ADIB (Adaptive Information Delivery). We thank Andreas Pfau for implementing the system.

References

[1] A. Dengel. Six thousand words about multi-perspective personal document management. In *Proc. EDM, IEEE Int. Workshop on the Electronic Document Management in an Enterprise Computing Environment, Hong Kong, China*. IEEE Computer Society, 2006.

[2] B. Horak. ConTag - A Tagging System linking the Semantic Desktop with Web 2.0. Diploma thesis, University Kaiserslautern, August 2006.

[3] M. Kiesel. Kaukolu: Hub of the semantic corporate intranet. In *SemWiki Workshop, ESWC 2006*, pages 31–42, 2006.

[4] H. Maus, H. Holz, A. Bernardi, and O. Rostanin. Leveraging Passive Paper Piles to Active Objects in Personal Knowledge Spaces. In *Professional Knowledge Management. Third Biennial Conference, WM 2005, Kaiserslautern, Germany, April 2005. Revised Selected Papers*, volume 3782 of *LNAI*, pages 50–59. Springer, 2005.

[5] L. Sauermann, A. Bernardi, and A. Dengel. Overview and Outlook on the Semantic Desktop. In S. Decker, J. Park, D. Quan, and L. Sauermann, editors, *Proc. of the First Semantic Desktop Workshop at the ISWC Conference 2005*, 2005.

Gestural Interaction for an Automatic Document Capture System

Christian Kofler, Daniel Keysers
German Research Center for Artificial Intelligence (DFKI),
Kaiserslautern, Germany
{christian.kofler, daniel.keysers}@dfki.de

Andres Koetsier, Jasper Laagland
University of Twente, Enschede, The Netherlands
{a.koetsier-1, j.laagland}@student.utwente.nl

Thomas M. Breuel
Technical University of Kaiserslautern, Germany
tmb@informatik.uni-kl.de

Abstract

The amount of printed documents used today is still very large despite increased use of digital formats. To bridge the gap between analog paper and digital media, paper documents need to be captured. We present a prototype that allows for cost-effective, fast, and robust document capture using a standard consumer camera. The user's physical desktop is continuously monitored. Whenever a document is detected, the system acquires its content in one of two ways. Either the entire document is captured or a region of interest is extracted, which the user can specify easily by pointing at it. In both modes a high resolution image is taken and the contained information is digitized. The main challenges in designing and implementing such a capturing system are real-time performance, accurate detection of documents, reliable detection of the user's hand and robustness against perturbations such as lighting changes and shadows. This paper presents approaches that address these challenges and discusses the integration into a robust document capture system with gestural interaction.

1. Introduction

In 1975, the *Business Week* confidently foresaw the paperless office to be close [1] but still this vision has not become reality. Instead, the use of paper in a typical office doubled since then from 100 to 200 pounds of paper per head [2]. Although the digital alternatives for mail, news and other forms of information are mature, the analog versions are still widely used and will continue to play an important role not only in office life [10]. Instead of ignoring the information available in paper form, easy transformation into the digital world is required to manage, archive and share it using the advantages of modern electronic communication.

Today, there are several ways of performing such a transformation of information from a document which is available on paper into a digital version of it. The user can e.g. use a scanner or a digital camera or even her mobile phone to take a picture of that document. Depending on the intended use of the information to be digitized there are subsequent steps to be taken, such as cropping the image to the exact boundaries of the entire document or a subregion of interest and extracting textual information by employing Optical Character Recognition (OCR). Many users regard these steps as obstacles and still prefer to transcribe the parts of the document they want to have available in digital form.

The system we present here removes these obstacles and supports the user digitizing documents either in oblivious or in interactive mode. A first version of the oblivious mode of the system was presented in [7]. Since then, we have improved the system in terms of performance, accuracy and robustness but the concept is still the same: The user works on his desk and is oblivious of the document capture system which continuously captures all new documents which appear on the users workspace. The textual content of these documents is then made accessible via a full-text search engine. This mode works completely without physical user interaction, and therefore without interrupting the user's everyday work-flow.

In discussions with users of the system it was frequently suggested that a possibility to select a region of interest within a document would be beneficial. Hence, we started to integrate gesture recognition into the document capture system as a comfortable way to interact with it. In the interactive mode of our current prototype the user can point at the document to define a region he is interested in. The text line which is closest to this point will be detected and the textual content extracted.

In both modes, oblivious and interactive, we employ our open-source OCR system OCRopus[1] to extract text from the document image.

The central component of the presented system is the detection of a document when it is placed in the viewfinder zone of the input device, i.e. a standard consumer camera. After the exact detection of the boundaries of the document a high-resolution image is taken and the document image is extracted. The perspective distortion is then removed from the captured document image to gain a rectified version of it. If the interactive mode is enabled the high-resolution of the placed document is only taken if the user pointed at a region of interest within it. Then the captured document is annotated with the pointing location which can be reused later-on in further processing steps.

2. System Design and Implementation

The current prototype of the presented system is intended to consistently visualize bridging the gap between physical and digital desktop. Hence, the demonstrator is a special wooden desk on top of which a standard consumer camera can be mounted. The only requirement for the employed camera is support for the standardized Picture Transfer Protocol (PTP) developed by the International Imaging Industry Association to allow the transfer of images from digital cameras to computers without the need of additional device drivers. A number of tests with commonly used cameras showed that only few manufacturers incorporate reasonable PTP support in their cameras. Currently we are using a 7 mega-pixel Canon A 620 camera.

The viewfinder image of the camera is requested continuously and handled in a sequence of processing steps. First, the parts of the image which constitute background are detected and masked to reduce data for all subsequent steps. The remaining pixels are considered foreground. An important step to achieve higher accuracy and robustness is the detection of shadows which are cast by foreground objects, such as the hand of the user. The main component of the system is the document detection. If fingertip detection is enabled, the respective processing steps are performed to determine the point of interest in the document at which the user pointed with his index finger.

Figure 1 shows the demonstrator hardware and Figure 2 gives an overview of the software components involved in the presented system. Each of the processing steps is discussed in more detail in the following sections.

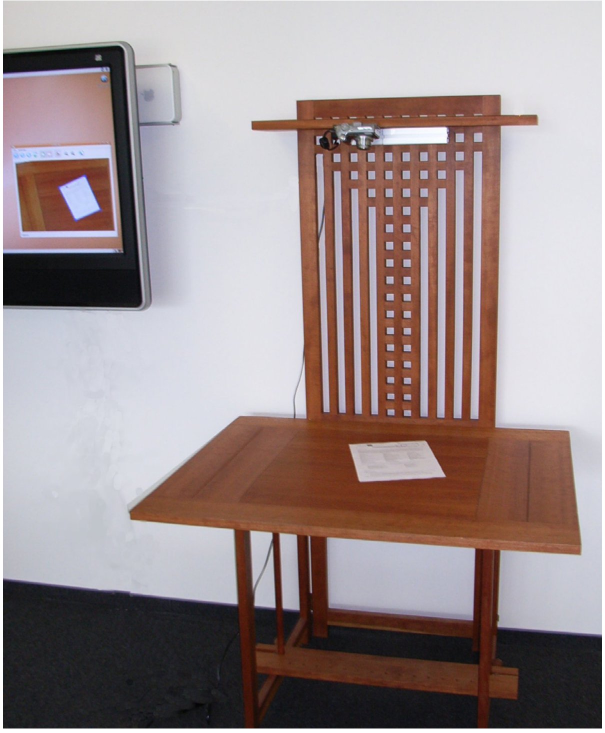


Figure 1. The demonstrator of the presented document capture system.

2.1. Background Detection

In order to reduce the amount of data for the subsequent processing steps a distinction between background and foreground of the current input image is necessary. Therefore, a dynamic background model is initialized as soon as the system starts to operate and updated continuously to be robust against local changes, such as moving background objects, and global changes, such as changing lighting conditions. Additional sources of global change are the automatic white balancing and color adaptation of the used consumer camera.

Based on the dynamic model, background subtraction is performed on each input image to determine foreground regions. An overview of dynamic background subtraction strategies can be found in [9]. As the requirements for our interactive system include real-time tracking with reasonable accuracy we decided to use a mixture of Gaussians for robust background subtraction. This method is described in [11] where it is used for tracking traffic.

A pixel X is defined in the RGB color space as $X = \{R, G, B\}$. The history of a pixel $\{X_1, ..., X_t\}$ is modeled

[1] http://www.ocropus.org

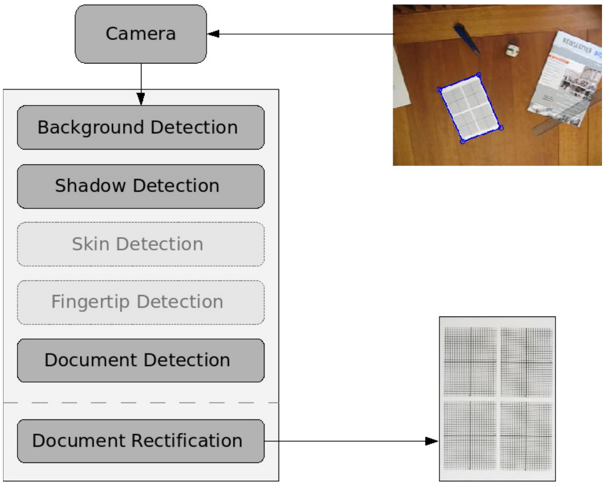


Figure 2. The components of the presented system: The viewfinder image of the camera is processed with background, shadow and document detection, optionally skin and fingertip detection is performed. The detected document is then extracted by taking a high resolution image and rectifying the respective image part.

by K Gaussian densities. The probability of observing pixel X_t is:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t}\, \mathcal{N}(X_t, \mu_{i,t}, \Sigma_{i,t})$$

where K is the number of Gaussian densities, $\omega_{i,t}$ is the weight of the i-th density at time t, $\mu_{i,t}$ is the mean of the i-th density at time t, $\sum_{i,t}$ is the covariance matrix of the i-th density at time t and $\mathcal{N}$ is the Gaussian probability density function:

$$\mathcal{N}(X, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}}\, e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)}$$

Here, we use diagonal covariance matrices of the form $\Sigma_{i,t} = \sigma^2_{i,t} I$

The K Gaussian densities are sorted according to their weights in descending order. We used the first N of the K Gaussians to define background, following a suggestion made in [12]. This means that the N Gaussian with the highest weights are considered background. Whenever a new pixel is presented to the model, the model is updated by first iterating through the K Gaussians and determining the density that best explains the pixel value and then updating the weights of all densities.

If none of the K densities explains the pixel value sufficiently well, the density with the lowest weight (i.e. the K-th density) is replaced by a new density with $\mu_{K,t} = X_t$ and low $\sigma_{K,t}$ and $\omega_{K,t}$.

The weights are updated according to

$$\omega_{i,t} \leftarrow (1-\alpha)\omega_{i,t-1} + \alpha(M_{i,t})$$

where α is the learning rate and $M_{i,t}$ is 1 for the density i that matched X_t and 0 for the other densities.

The remaining parameters are updated according to

$$\begin{aligned}
\mu_{i,t} &\leftarrow (1-\rho)\mu_{i,t-1} + \rho X_t \\
\sigma^2_{i,t} &\leftarrow (1-\rho)\sigma^2{}_{i,t-1} + \rho(X_t - \mu_{i,t})^T (X_t - \mu_{i,t}) \\
\rho &= \alpha\, \mathcal{N}(X_t, \mu_{i,t}, \sigma_{i,t})
\end{aligned}$$

The advantage of using a mixture of Gaussians is that objects that are new in the scene can quickly be merged with the background and changes in light intensity can quickly be resolved. One disadvantage is that whenever a foreground object (e.g. a hand or a document) remains at the same position long enough it will dissolve into the background. To avoid this problem we add masks to the background model. In the mask, a value of 1 indicates that the corresponding pixel is to be processed as foreground and not updated, while a 0 indicates that the pixel is background and should be updated. This requires fast detection of hands and documents because otherwise they will be dissolved before they are detected.

2.2. Shadow Detection

Whenever a user points at a document, the arm and hand create a drop shadow that the chosen background model does not take into account. Hence, drop shadows will be considered foreground, complicating the exact detection of the user's fingertip. To eliminate this problem we model drop shadows and exclude them from the foreground. Detected shadows are not included in the background mask as experiments showed that separate masking of shadow results in better performance.

We use a method similar to the approach presented in [5]. Each new pixel that is presented to the background model is first checked with the current background Gaussians. If the pixel is considered to be a shadowed value of one of the foreground Gaussians, the pixel is labeled accordingly. The method used for detecting shadows is based on [3].

The goal of the shadow detection in our system is to reduce the foreground to the actual objects that should be detected in a scene. Figure 3 illustrates how the employed algorithm can accomplish this task.

2.3. Skin Detection

Whenever a new object appears in the viewfinder of the camera the respective region will be considered background

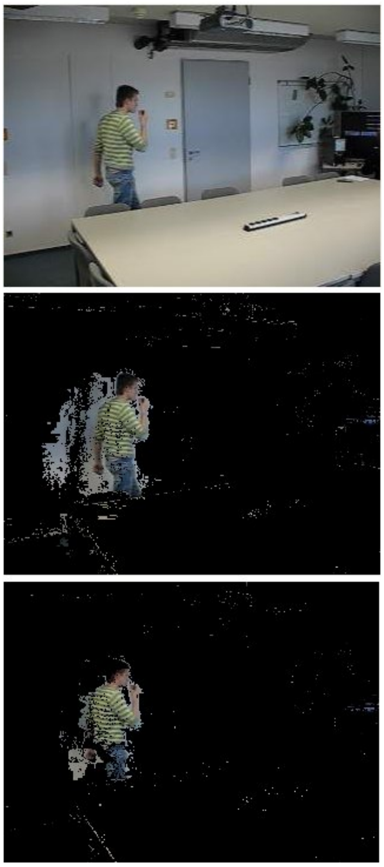


Figure 3. Detection of foreground in an input image (top) using only background detection (center) and background detection in combination with shadow detection (bottom). Only non-black pixels are considered foreground.

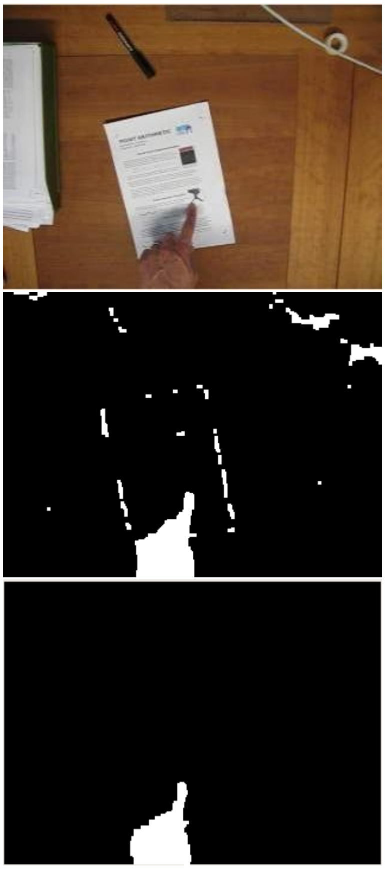


Figure 4. Segmentation of the hand in an input image (top) using only skin-color (center) and skin-color in combination with background detection (bottom).

after a number of frames. In this restricted amount of time, the system has to distinguish hand from non-hand candidate regions. This is done by detecting skin-colored regions. There exist several alternative approaches to detect skin; an overview can be found in [13].

We are using a Bayes classifier with skin probability map as proposed by Jones et al. [4]. The classification of the two sets of pixels involves a histogram which is based on the Compaq Cambridge Research Lab image-database.

An rgb pixel value is then classified according to the ratio $P(\text{rgb}|\text{skin})/P(\text{rgb}|\neg\text{skin})$ obtained from the model, which is compared to a threshold value.

Figure 4 shows the detection of skin in an input image and how the combination of skin and background detection improves accuracy.

2.4. Fingertip Detection

After segmenting foreground from background and detecting skin-colored regions, a matching algorithm verifies the presence of a hand in the segmented foreground. The fast normalized cross correlation [8] is used here.

Once a hand is detected, the exact location of the pointing finger needs to be identified. As the user is usually interacting with the system in front of the desk, the assumption is made that he will always point ‘upwards’, i.e. away from his body. Hence, the tip of the pointing finger will always be at the highest y-value of the detected hand region. While the image is flood-filled to create a mask for the background subtraction locating the finger tip is done in the same iteration. Based on informal experiments, we examine the eight top-most rows of pixels of the hand region. If their widths are within suitable thresholds the middle of the detected fingertip is considered the region of interest of the user as shown in Figure 5.

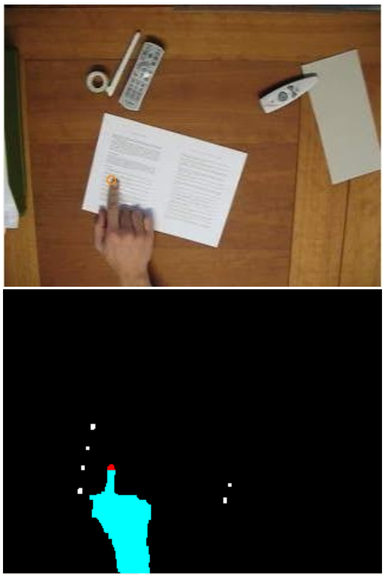
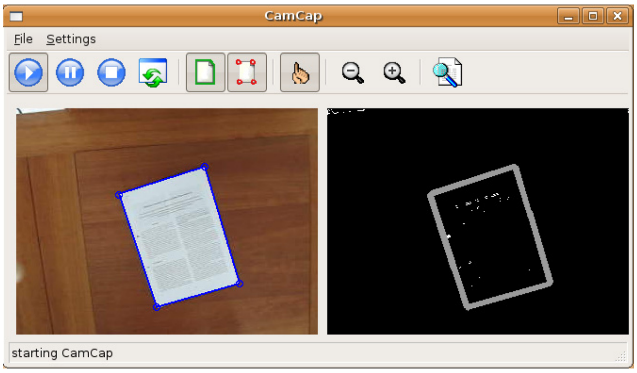


Figure 5. Example of detecting the hand and the pointing finger tip.



starting CamCap

Figure 6. Example of the demonstrator GUI with debug-output to visualize the detection a document by finding gradients in the foreground.

2.5. Document Detection

The first step in capturing a document is to know that a document actually is present in the viewfinder image. To detect a document a number of methods are available including background color differencing and background gradient differencing. Background color differencing compares each pixel color in the viewfinder image to the same pixel in the background image in RGB space. The color difference is the Euclidean distance between the two pixel colors. When the color difference exceeds a certain threshold the pixel is considered to be foreground. Although this approach is intuitive and easy to implement it has a few drawbacks. The first drawback is the sensitivity to noise. If a shadow or a highlight appears in the viewfinder image the pixels inside this noisy region are falsely classified as foreground pixels. Another problem is the fact that, when a document is placed in the viewfinder image of the camera, the lighting will change, causing the camera to adjust its white balance and exposure settings. This will cause a global color change in the viewfinder image which could result in the entire image appearing as foreground.

Because of these problems, another approach was used for detecting possible foreground objects in the viewfinder image. This second approach also uses differencing of the current image and the background image. However, it does not use color information directly to classify a pixel as foreground or background. Instead, it first creates a gradient image of both the background and the viewfinder image using a simple Sobel kernel (-1,0,1) in both horizontal and vertical direction. The next step is subtracting each gradient pixel of the background from the gradient pixels in the viewfinder image. The result is an image with gradients which only appear in the viewfinder image and not in the background image as shown in Figure 6. Gradients that only exist in the background image may be caused by objects being removed from the desk and become negative gradients in the difference image. The difference image is then thresholded to only keep strong positive gradients. The advantage over color differencing is that global color changes do not influence the gradient images and thus will not cause the entire image to be classified as foreground. Also shadows and highlights are removed because they rarely contain any sharp edges. The gradient image now contains all the lines of the foreground objects in the viewfinder image. These objects do not always have to be documents but can also be other items placed on the desk. To reduce the candidate lines to straight lines which could constitute the edges of a document a Hough transform is performed. The result is a set of straight lines, each represented by a start and end point. These lines are then clustered so double lines and close lines are removed. If a document exists in the set of lines it can be assumed that a document is surrounded by four lines. In order to find enclosed areas, we iterate through the set of lines matching end points of one line with the start point of another line. When an end point of a certain line reaches the start point of the first line the algorithm returns the area enclosed by these lines as a document.

2.6. Document Rectification

For each document detected in the viewfinder image a set of corner-points is saved according to [14]. Next, the document detector acquires a high resolution image from the

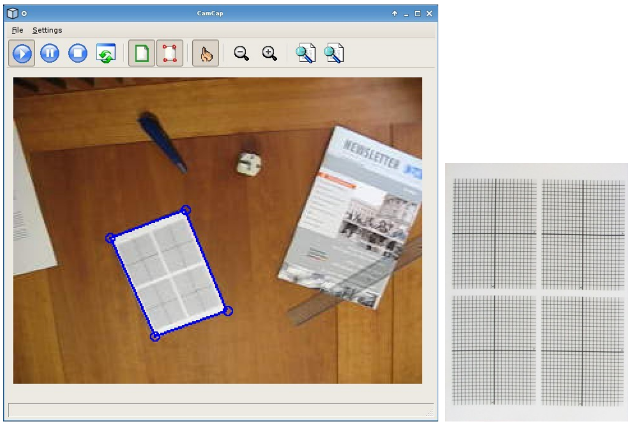


Figure 7. Detected document in the GUI of the system and the high-resolution document image after extraction and rectification.

camera on which the actual document rectification is performed. An example document rectification can be seen in Figure 7. The coordinates of the corner-points of a document are scaled to match the same points in the high resolution image. By scaling these points precision can be lost due to rounding errors or viewfinder deviation with respect to the high resolution image. In order to (re)gain accuracy the document detector will try to detect the exact position of the document corners in the high resolution image. This is done by extracting a patch from the high resolution image for each corner-point. The center of each patch is the scaled location of the corner-point from the original viewfinder image. In this patch edges and lines are detected by using the same methods as for document detection. The crossings of all the lines are calculated and the point with most crossings is considered to be an exact corner-point of the document. This processing step results in less noise and parts of the background at the borders of the captured document. After the detection of the corner-points the algorithm rectifies the image to eliminate distortion caused by projection. For rectification of the document the method in [14] is used which is based on the standard pinhole model to calculate the projection from a space point M to an image point m as shown in Figure 8.

3. Application

The system we present in this paper can serve as a basis for various applications based on the rectified image of a document along with the coordinates of a certain region of interest. The most obvious next step is to find the exact text line at which the user pointed and perform OCR on it. Another use case of the interactive mode of the presented system is illustrated in Figure 9: The user points at a reference in a bibliography. First, the paragraph constituting the reference of interest is identified with a layout analysis algorithm. The contained text is then obtained from the respective part of the input image by performing OCR on it. After that, the extracted text of the reference is forwarded to a system that extracts bibliographic meta-data from scientific references[2] [6]. With this information in BibTeX format, the respective scientific paper can be easily found in digital libraries, such as CiteSeer[3] or CiteULike[4].

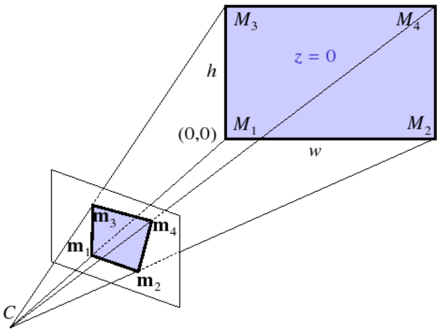
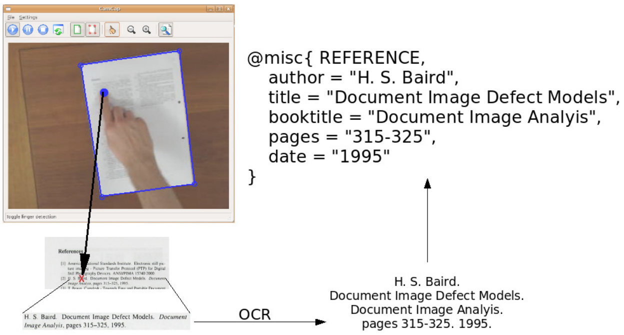


Figure 8. Conversion from image to space points based on the pinhole model.



```
@misc{ REFERENCE,
   author = "H. S. Baird",
   title = "Document Image Defect Models",
   booktitle = "Document Image Analyis",
   pages = "315-325",
   date = "1995"
}
```

H. S. Baird.
Document Image Defect Models.
Document Image Analyis.
pages 315-325. 1995.

Figure 9. Example for the extraction of bibliographic meta-data: The user points at a reference, a high resolution image is taken and the correpsponding image part is extracted based on the location of the fingertip of the user. Then OCR is performed on the subimage and the recognized text is processed to obtain a BibTeX representation of the reference.

[2] http://demo.iupr.org/refrec/
[3] http://citeseer.ist.psu.edu/
[4] http://www.citeulike.org/

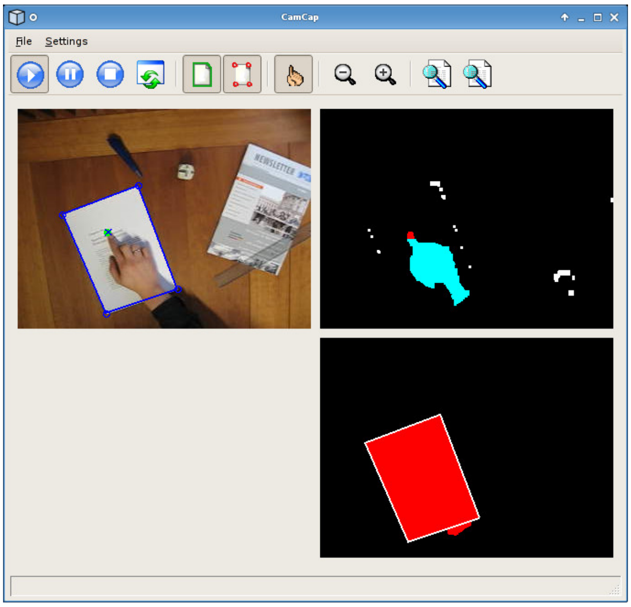


Figure 10. Example of the demonstrator GUI with debug-outputs to visualize the detection of hand/fingertip and document.

4. Summary

We have presented the status of our work on document capture systems. The current prototype supports two modes, i.e. oblivious and interactive capture. In the oblivious mode the user works at his desk as usual and all paper documents that entered the field of view of the mounted camera are archived digitally. In the interactive mode the user can specify a region of interest within a document by pointing at it. The whole document is captured and the region of interest is made accessible immediately.

The system offers a fast, robust and cost-effective way of capturing documents on the physical desktop. The accurate detection allows for real-world applications that bridge the gap to the virtual desktop. Figure 10 shows another example of the demonstrator application in a debug mode that shows the detection of a document and the fingertip of the user.

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).

References

[1] The office of the future. *Business Week*, 2387:48 – 70, 30 June 1975.

[2] J. S. Brown and P. Duguid. *The Social Life of Information*. Harvard Business School Press, Boston, MA, USA, 2002.

[3] T. Horprasert, D. Harwood, and L. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings IEEE ICCV '99 FRAME-RATE Workshop*, 1999.

[4] M. Jones and J. Rehg. Statistical color models with application to skin detection. In *International Journal of Computer Vision*, volume 46, pages 81 – 96, 2002.

[5] P. KaewTrakulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings 2nd European Workshop on Advanced Video-based Surveillance Systems (AVBS01)*, Kingston upon Thames, September 2001.

[6] M. Kraemer, H. Kaprykowsky, D. Keysers, and T. Breuel. Bibliographic meta-data extraction using probabilistic finite state transducers. In *Proceedings of ICDAR 2007*, in press.

[7] C. H. Lampert, T. Braun, A. Ulges, D. Keysers, and T. M. Breuel. Oblivious document capture and real-time retrieval. In *International Workshop on Camera Based Document Analysis and Recognition (CBDAR)*, pages 79–86, Seoul, South Korea, aug 2005.

[8] J. Lewis. Fast normalized cross-correlation. In *Vision Interface*, pages 120 – 123, 1995.

[9] M. Piccardi. Background subtraction techniques: a review. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2004)*, pages 3099 – 3104, The Hague, Netherlands, October 2004.

[10] A. J. Sellen and R. H. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.

[11] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 1999)*, volume 2, pages 244 – 252, 1999.

[12] C. Stauffer and W. Grimson. Learning patterns of activity using real-time trackingg. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:747 – 757, 2000.

[13] V. Vezhnevets, V. Sazonov, and V. Andreeva. A survey on pixel-based skin color detection techniques. In *Proceedings of the Graphicon*, pages 85 – 92, 2003.

[14] Z. Zhang and L. He. Whiteboard scanning and image enhancement. Technical report, MSR-TR-2003-39, June 2003.

Real-Time Document Image Retrieval with More Time and Memory Efficient LLAH

Tomohiro Nakai, Koichi Kise, Masakazu Iwamura
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Naka, Sakai, Osaka, 599-8531 Japan
nakai@m.cs.osakafu-u.ac.jp, {kise, masa}@cs.osakafu-u.ac.jp

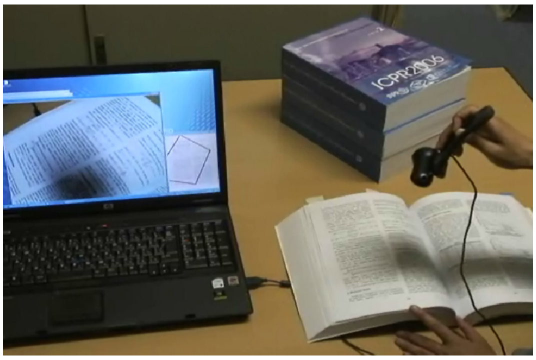


Figure 1. A scene of the demonstration.

We propose a real-time document image retrieval system using a web camera. This is an implementation of the camera based document image retrieval method presented in CBDAR07 oral session [1] as a real-time document image retrieval system. As shown in Fig. 1, the user can retrieve document images from a database by capturing paper documents with a web camera.

The overview of the proposed system is shown in Fig. 2. Firstly, feature points are extracted from a captured image using a web camera. Based on correspondences of feature points, a retrieval result is calculated. The retrieval result and captured region are presented to the user.

The proposed system has following two features. (1) Since the system use LLAH [1], it is robust to various types of disturbances. As shown in Fig. 3, the proposed system is robust to rotation, scaling, perspective distortion, occlusion and curvature. (2) Fast retrieval is realized even on a large scale database. In concrete terms, processing time is about 0.1 second per one query image on a database with 10,000 pages.

A sample program of the proposed system is downloadable from [2].

References

[1] T. Nakai, K. Kise and M. Iwamura, "Camera based document image retrieval with more time and memory efficient LLAH", Proc. CBDAR07, 2007 [to appear].

[2] `http://www.imlab.jp/LLAH/` .

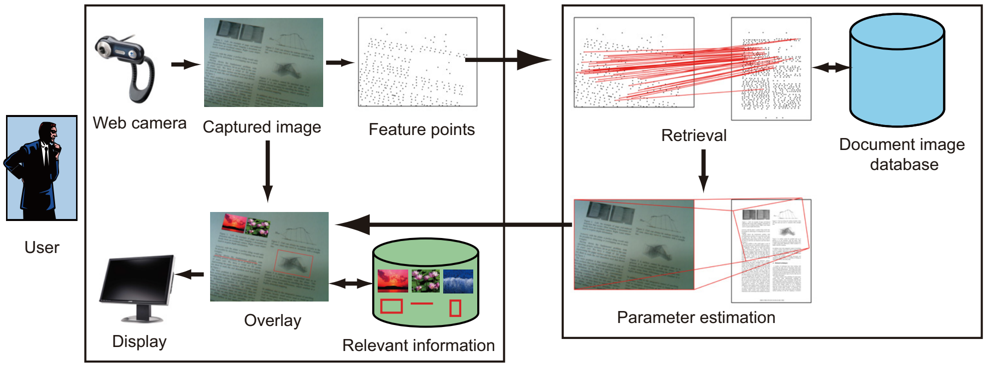
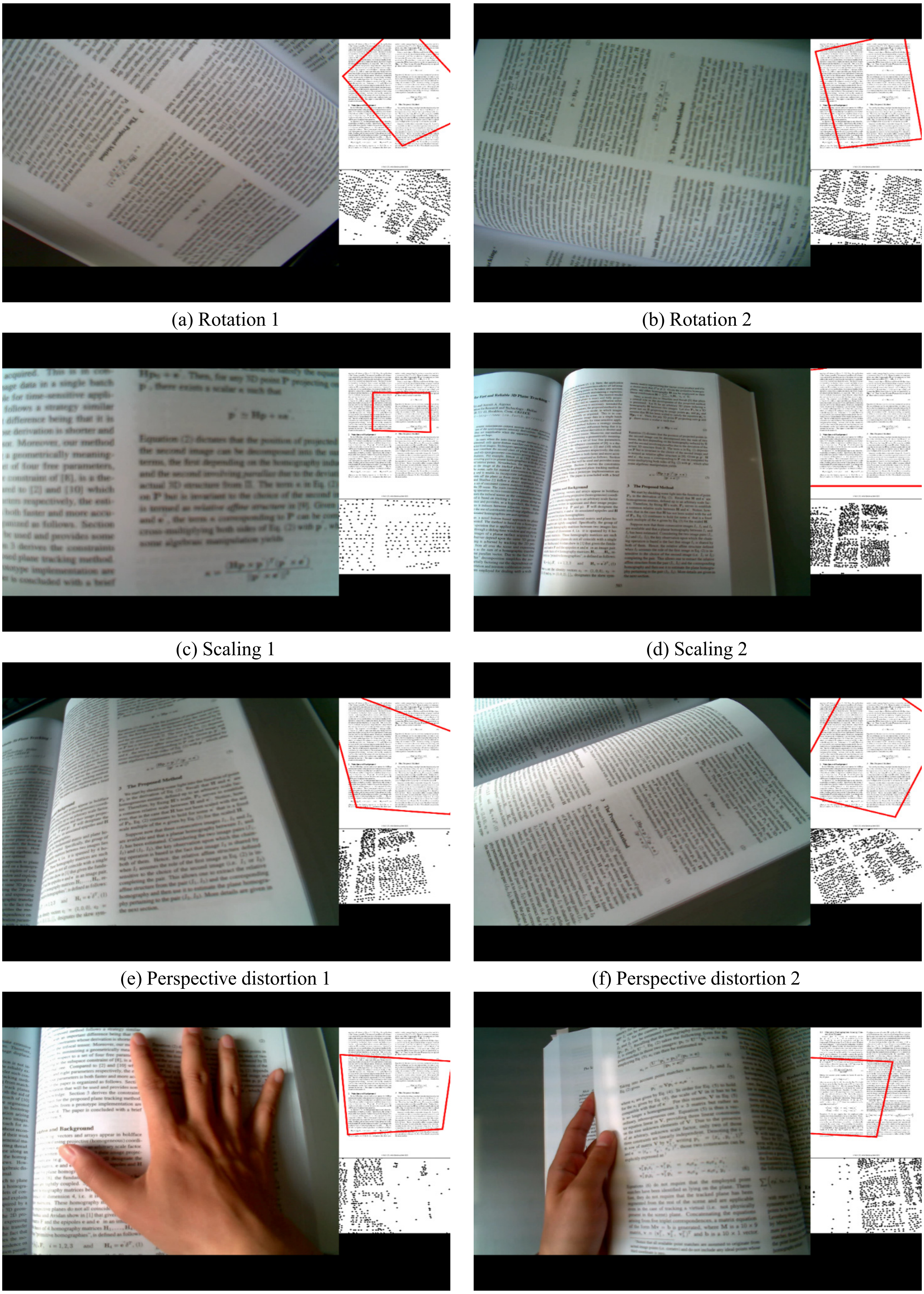


Figure 2. The overview of the proposed system.



(a) Rotation 1

(b) Rotation 2

(c) Scaling 1

(d) Scaling 2

(e) Perspective distortion 1

(f) Perspective distortion 2

(g) Occlusion

(h) Curvature

Figure 3. Robustness against various types of disturbances. The left part of each figure shows a captured image. The upper right part shows a retrieval result in which the red rectangle indicates the captured region. The lower right part shows feature points extracted from the captured image.

Demo: Applications of Document Analysis on Camera Phone

Xu Liu
Institute for Advanced Computer Studies
University of Maryland
liuxu@cs.umd.edu

Huiping Li
Applied Media Analysis, Inc. [1]
huiping@mobileama.com

Abstract

Camera phones have penetrated every corner of society. The combined image acquisition, processing, storage and communication capabilities in mobile phones have rekindled researchers' interests in applying pattern recognition and computer vision algorithms on camera phones in the pursuit of new mobile applications. This demonstration will highlight recent research on camera phone applications which enable a camera phone to:

- *Read barcodes (1D, 2D and video barcodes)*
- *Retrieve document from a one snapshot*
- *Recognize currency for the visually impaired*

We have addressed many of the challenges encountered in implementing vision and recognition algorithms on light-weight systems with embedded cameras. Often solutions require a fundamentally different approach than traditional document analysis techniques.

[1] Applied Media Analysis (AMA) Inc. is the leading provider of Mobile vision solutions for a variety of vertical and consumer markets that include healthcare, defense and consumers. Based upon our patent-ready mobile vision technology, a computer vision technology which allows camera-enabled handheld devices, such as PDAs and Smartphones, to read and see, AMA products turn users' camera phones into personal data scanners. By leveraging the convergence of sensor, processing and networking capabilities on commodity hardware, MobileVision creates an opportunity to build applications which link the existing physical world - traditionally processed primarily by the human visual system or dedicated hardware - to mobile content. In its rudimentary form, this platform enables devices to recognize barcodes and other language characters, such as the information printed on a business card.

Video Mosaicing for Document Imaging

Noboru Nakajima[†] Akihiko Iketani[†] Tomokazu Sato[†‡]
Sei Ikeda[‡] Masayuki Kanbara[†‡] Naokazu Yokoya[†‡]
†Common Platform Software Research Laboratories, NEC Corporation,
8916-47 Takayama, Ikoma 630-0101, Japan
‡Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma 630-0192, Japan
{n-nakajima@ay,iketani@cp}.jp.nec.com, {tomoka-s,sei-i,kanbara,yokoya}@is.aist-nara.ac.jp

Abstract

This paper describes a real-time video mosaicing method for digitizing paper documents with mobile and low-resolution video cameras. Our proposed method is based on the structure-from-motion technique with real-time processing. Therefore, it can dewarp an image suffering from perspective deformation caused by a slanted and curved document surface. The curved surfaces are reconstructed as an expanded flat image by 3D polynomial surface regression and projection onto a flat mosaic plane. The order of the polynomial is automatically determined according to the geometric AIC. Moreover, the real-time scanning and reconstruction process offers interactive user instruction on the style of user's camera handling, such as appropriateness of camera posture and scanning speed. Manual camera scanning for video mosaicing captures a point on the target multiple times and that allows other merits. They include super-resolution and shading correction. The experiments demonstrated the feasibility of our approach.

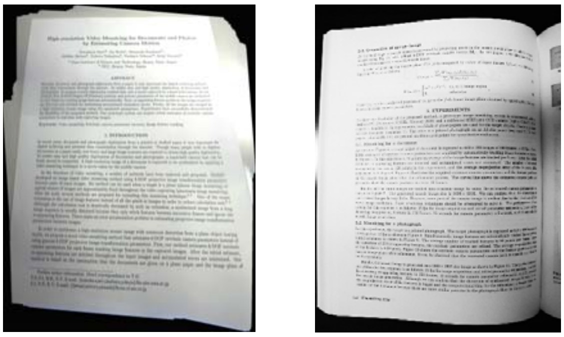
1. Introduction

Digital camera devices are getting increasingly commoditizing. But in contrast to their facility, their quality and assurance of document digitization are not sufficient for document analysis and recognition. The literature in the DAR field describes several technical challenges for their practical realization such as the followings [1-3]:

(1) dissolution for deficiency in resolving power,
(2) dewarping of curved surface suffering perspective deformation,
(3) elimination of illumination irregularity,
(4) target localization and segmentation.

A full A4 page image at 400 dpi requires more than 14.7M pixels, whereas a flatbed scanner enables a few thousands dpi. For the resolution deficiency problem (1), video mosaicing is one of the most promising solutions. In video mosaicing, partial images of a document are captured as a video sequence, and multiple frames are stitched seamlessly into one large, high-resolution image. Conventional video mosaicing methods usually achieve pair-wise registration between two successive images, and construct a mosaic image projecting all the images onto a reference frame, in general, the first frame. Szeliski [4] has developed a homography-based method, which adopts 8-DOF planar projective transformation. After his work, various extensions have been proposed [7-10]. One of the major developments is the use of image features instead of all of the pixels for reducing computational cost [5-6]. All of these methods, however, align the input images to the reference frame, and thus will suffer perspective slanted effect if the reference frame is not accurately parallel to the target surface, even though a target page is rigidly planar, as shown in Figure 1(a).

There actually exists a severe image distortion (2) when capturing a curved target, as shown in Figure 1(b). The homography-based methods are applicable just when a target is a piecewise planar, or the optical



(a) Mosaic image generated by conventional methods

(b) Curved page captured with a digital camera

Figure 1. Problems in digitized document image.

center of the camera is approximately fixed (panoramic mosaicing). Thus, if the target is curved, the above assumption no longer holds, and misalignment of images will degrade the resultant image. Although there are some video mosaicing methods that can deal with a curved surface, they require an active camera and a slit light projection device [10].

On the other hand, the usability of the system is also an important factor. Unfortunately, conventional video mosaicing methods have not focused on the usability of the system because interactive video mosaicing systems have not been developed yet.

In order to solve these problems, we employ a structure-from-motion technique with real-time processing. Our method is constructed of two stages as shown in Figure 2. In the first stage, a user captures a target document using a web-cam attached to a mobile computer (Figure 3). In this stage, camera parameters are automatically estimated in real-time and the user is supported interactively through a preview window for displaying a mosaic image on the mobile computer. After that, camera parameters are refined, and a high-resolution and distortion-free mosaic image is generated in a few minutes. The main contributions of our work can be summarized as follows: (a) perspective deformation correction from a mosaic image for a flat target, (b) dewarped mosaic image generation for a curved document, and (c) interactive user support on a mobile system. Camera scanning allows surplus textural information on the target, or a point on the document is captured multiple times in the image sequence. Other benefits to be reaped from our system include (d) super-high-definite image construction by super-resolution [11] for enhancing the solution for the low-resolution problem (1), and (e) shading correction by intrinsic image generation [12] for irregular illumination (3). In the objective of document capturing, the location of the target, at which a user principally points a camera, is straightforward here for the target location problem (4).

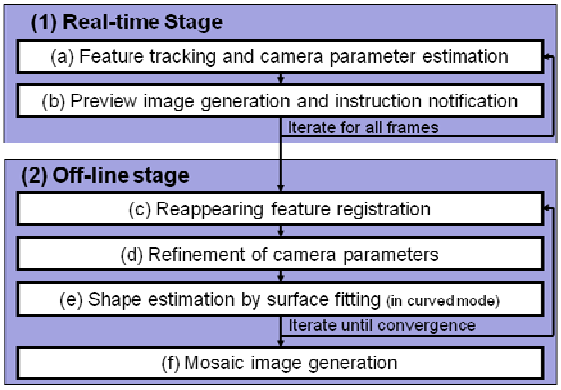

(1) Real-time Stage
(a) Feature tracking and camera parameter estimation
(b) Preview image generation and instruction notification
Iterate for all frames
(2) Off-line stage
(c) Reappearing feature registration
(d) Refinement of camera parameters
(e) Shape estimation by surface fitting (in curved mode)
Iterate until convergence
(f) Mosaic image generation

Figure 2. Flow of proposed method.

Figure 3. Mobile PC with web-cam.

The assumptions made in the proposed method are that intrinsic camera parameters, including lens distortion, are fixed and calibrated in advance. For curved documents, the curvature must lie along a one-dimensional direction, or be cylindrical, and varies smoothly on a page.

2. Video Mosaicing for Flat and Curved Document

This section describes a method for generating a high-resolution and distortion-free mosaic image from a video sequence. The flow of the proposed method is given in Figure 2. Although our system has two modes, one for a flat target (flat mode) and the other for a curved target (curved mode), the basic flow for these two modes is almost the same. In the real-time stage, the system carries out 3D reconstruction processes frame by frame by tracking image features (a). The preview of a temporal mosaic image is rendered in real-time and updated in every frame to show what part of the document has already been captured (b). After the real-time stage, the off-line stage is automatically started. In this stage, first, re-appearing image features are detected in the stored video sequence (c), and camera parameters and 3D positions of features estimated in the real-time stage are refined (d). If the system works in the curved mode, surface parameters are also estimated by fitting a 3D surface to an estimated 3D point cloud (e). After several iterations, a distortion-free mosaic image of high-resolution is finally generated (f) applying the super-resolution method. In the following sections, first, extrinsic camera parameters and an error function used in the proposed method are defined. Stages (1) and (2) are then described in detail.

2.1. Definition of Extrinsic Camera Parameter and Error Function

In this section, the coordinate system and the error function used for 3D reconstruction are defined. We define the coordinate system such that an arbitrary point $\mathbf{S}_p = (x_p, y_p, z_p)$ in the world coordinate system is projected to the coordinate $\mathbf{x}_{fp} = (u_{fp}, v_{fp})$ in the *f*-th image plane. Defining 6-DOF extrinsic camera parameters of the *f*-th image as a 3x4 matrix $\mathbf{M}_f$ the relationship between 3D coordinate $\mathbf{S}_p$ and 2D coordinate $\mathbf{x}_{fp}$ is expressed as follows:

$$(au_{fp}, av_{fp}, a)^T = \mathbf{M}_f (x_p, y_p, z_p, 1)^T , \qquad (1)$$

where *a* is a normalizing parameter. In the above expression, $\mathbf{x}_{fp}$ is regarded as a coordinate on the ideal camera with the focal length of 1 and of eliminated radial distortion induced by the lens. In practice, however, $\mathbf{x}_{fp}$ is the projected position of $\hat{\mathbf{x}}_{fp} = (\hat{u}_{fp}, \hat{v}_{fp})$ in the real image, which is given transferring according to the known intrinsic camera parameters including focus, aspect, optical center and distortion parameters. This transformation from $\hat{\mathbf{x}}_{fp}$ to $\mathbf{x}_{fp}$ is applied only to coordinate calculation in the real-time stage regarding computational cost, whereas it will be considered precisely as a part of image dewarping in off-line stage, and omitted for simplicity in the rest of this paper.

Next, the error function used for 3D reconstruction is described. In general, the projected position $\mathbf{x}_{fp}$ of $\mathbf{S}_p$ to the *f*-th subimage frame does not coincide with the actually detected position $\mathbf{x}'_{fp} = (u'_{fp}, v'_{fp})$, due to errors in feature detection, extrinsic camera parameters and 3D feature positions. In this paper, the squared error E_{fp} is defined as an error function for the feature *p* in the *f*-th frame as follows:

$$E_{fp} = \left|\mathbf{x}_{fp} - \mathbf{x}'_{fp}\right|^2 . \qquad (2)$$

2.2. Real-time Stage for Image Acquisition

In the real-time stage, extrinsic camera parameters are estimated in real-time to show some information that helps users in image capturing. First, a user may very roughly set the image plane of the camera parallel to the target paper. Note that this setting is not laborious for users because it is done only for the first frame. This setting is used to calculate initial values for the real-time camera parameter estimation. The user then starts image capturing with free camera motion.

As shown in Figure 2, the real-time stage is constructed of two iterative processes for each frame. First, the extrinsic camera parameter is estimated by tracking features (a). A preview image of generating a mosaic image and instruction for controlling camera motion speed are then shown in the display on the mobile computer and updated every frame (b). The following describes each process of the real-time stage.

Step (a): Feature tracking and camera parameter estimation.

An iterative process to estimate the extrinsic camera parameters and 3D position $\mathbf{S}_p$ of each feature point is described. This process is an extension of the structure-from-motion method in [13].

In the first frame, assuming that the image plane in the first frame is approximately parallel to the target, rotation and translation components in $\mathbf{M}_f$ are set to an identity matrix and 0, respectively. For each feature point detected in the first frame, its 3D position $\mathbf{S}_p$ is set to $(u_{1p}, v_{1p}, 1)$, based on the same assumption. Note that these are only initial values, which will be corrected in the refinement process (Figure 2 (d)).

In the succeeding frames ($f > 1$), $\mathbf{M}_f$ is determined by iterating the following steps until the last frame.

Feature point tracking: All the image features are tracked from the previous frame to the current frame by using a standard template matching with Harris corner detector [14]. The RANSAC approach [15] is employed for eliminating outliers.

Extrinsic camera parameter estimation: Extrinsic camera parameter $\mathbf{M}_f$ is estimated using the feature position (u'_{fp}, v'_{fp}) and its corresponding 3D position $\mathbf{S}_p = (x_p, y_p, z_p)$. Here, extrinsic camera parameters are obtained minimizing the error summation $\sum_p E_{fp}$ w.r.t. 6-DOF elements in $\mathbf{M}_f$ by the Levenberg-Marquadt method. For 3D position $\mathbf{S}_p$ of the feature *p*, the estimated value in the previous iteration is used.

3D feature position estimation: For every feature point *p* in the current frame, its 3D position $\mathbf{S}_p = (x_p, y_p, z_p)$ is refined by minimizing the error function $\sum_{i=1}^{f} E_{ip}$. In the case of the flat mode, *z* value of the $\mathbf{S}_p$ is fixed to a constant to improve the accuracy of estimated parameters.

Addition and deletion of feature points: In order to obtain accurate estimates of camera parameters, stable features should be selected. The set of features to be tracked is updated based on the feature reliability [13].

Iterating the above steps, extrinsic camera parameters $\mathbf{M}_f$ and 3D feature positions $\mathbf{S}_p$ are estimated.

Step (b): Preview image generation and instruction notification for user.

Figure 4 shows displayed information for a user in our system. Usually, the user watches the right side window in the real-time stage. If necessary, the user can check the input subimages and estimated camera path on the left side windows. The preview image helps the user to confirm a lacking part of the target to be scanned.

For each frame, the captured subimage is resized to a lower resolution and stored to texture memory. Every stored texture is dewarped onto the mosaic image plane by texture mapping using Eq. (1). The boundary of the current frame image is colored in the preview so that the user can grasp what part of the target has currently been captured.

On the other hand, the instruction shown in the bottom of the right window in Figure 4 helps the user to control the speed of camera motion. In our system, there is appropriate speed for camera motion that is shown by an arrow mark in the speed gauge, and it is graded as three ranges; “too slow", “appropriate," and “too fast." Too fast camera motion causes a worse mosaic image quality because the error in camera parameter estimation increases due to tracking errors and shortage of tracking span of each feature. In contrast, too slow motion consumes computational resources redundantly because the number of images increases to capture the whole document. To obtain a better result, the user controls camera speed so as to keep the arrow mark being inside “appropriate" range. In turn, too much slanted camera posture, which is still tolerable for 3D reconstruction, can cause degradations such as irregular resolution and partial defocus blur in a mosaic image. The camera pose can also be checked in the left-bottom window.

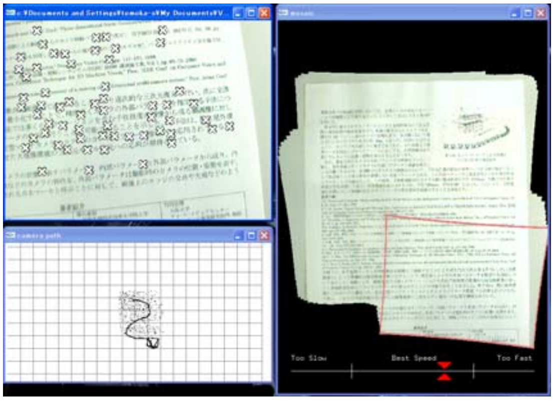
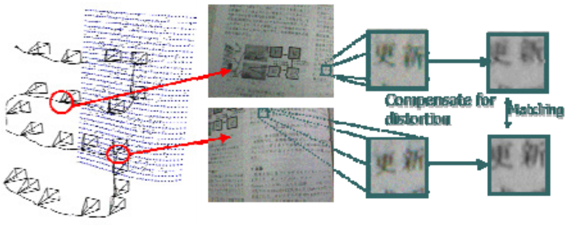


Figure 4. User Interface for real-time stage. Left-top: input image and tracking feature point. Left-bottom: temporarily estimated camera path and posture. Right: preview of ongoing mosaicing preview and speed indicator of camera motion.



(a) (b) (c) (d)

Figure 5. Registrations of reappearing features. (a) Camera path and 3D feature position. (b) Two temporally distant frames sharing part of scope. (c) Templates of the corresponding features. (d) Templates in which perspective distortion is corrected.

2.3. Off-line Stage for 3D Reconstruction Refinement and Mosaic Image Generation

This section describes the 3D reconstruction parameter refinement and shape regression, so as to minimize the summation of the error function all over the input sequence and feature misalignment from a parameterized columnar polynomial surface. First, as shown in Figure 2, feature points that reappear in temporally distant frames are homologized (c). Using these reappearing features, the 3D reconstruction is refined (d). Then, by fitting a curved surface to 3D feature points, the target shape is estimated (e). After a few iterations of steps (c) to (e), a dewarped mosaic image of the target is generated according to the target shape and the extrinsic camera parameters (f). Details of steps (c) to (f) are described below.

Step (c): Reappearing feature registration.

Due to the camera motion, the features relatively pass though the scope of the frames. Some features reappear in the sight after the previous flameout, as shown in Figure 5. This step detects these reappearing features, and distinct tracks belonging to the same reappearing feature are linked to form a single long track. This will give tighter constraints among the sequence of camera parameters in temporally distant frames, and thus makes it possible to reduce the cumulative errors caused by sequential camera scanning.

Reappearing features are detected by examining the similarity of the patterns among features belonging to distinct tracks. The problem here is that even if two patterns belong to the same feature on the target, they can have different appearance due to perspective distortion. To remove this effect, first, templates of all the features are projected to the fitted surface (described later). Next, feature pairs whose distance in 3D space is less than a given threshold are selected and tested with the normalized cross correlation function. If the correlation is higher than a certain threshold, the feature pair is regarded as reappearing features. Note that in the flat mode, the shape of the target is simply assumed as a plane. When the system works in the curved mode, this step is skipped at an initial iteration of steps (c) to (e) because surface parameters have not been estimated at the first iteration.

Step (d): Refinement of 3D reconstruction.

Since the 3D reconstruction process described in Section 2.2 is an iterative process, its result is subject to cumulative errors. In this method, by introducing a bundle-adjustment framework [16], the extrinsic camera parameters and 3D feature positions are globally optimized so as to minimize the sum of re-projection errors, which is given as follows:

$$E_{all} = \sum_f \sum_p E_{fp} \, . \tag{3}$$

As for reappearing features, all the tracks belonging to the same reappearing feature are linked, and treated as a single track. This enables the extrinsic camera parameters and 3D feature positions to be optimized, maintaining consistency among temporally distinct frames.

Step (e): Target shape estimation by surface fitting.

In this step, assuming the curvature of the target lies along one direction, the target shape is estimated using 3D point clouds optimized in the previous step (d). Note that this stage is skipped in the case of the flat mode. First, the principal curvature direction is computed from the 3D point clouds as shown in Figure 6. Next, the 3D position of each feature point is projected to a plane perpendicular to the direction of minimum principal curvatures. Finally, a polynomial equation of variable order is fitted to the projected 2D coordinates, and the target shape is estimated.

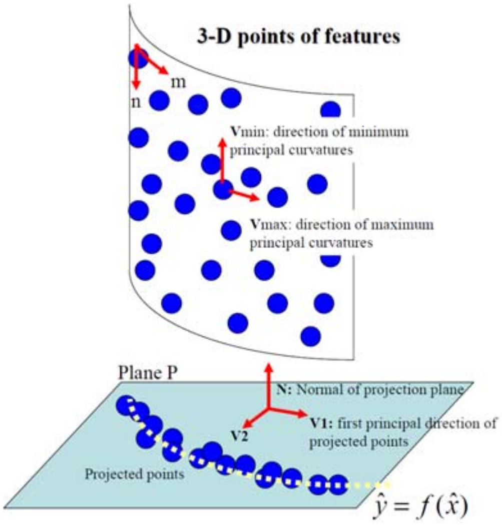

3-D points of features
m
n
Vmin: direction of minimum principal curvatures
Vmax: direction of maximum principal curvatures
Plane P
N: Normal of projection plane
V1: first principal direction of projected points
V2
Projected points
$\hat{y} = f(\hat{x})$

Figure 6. Polynomial surface regression for target shape approximation

Let us consider for each 3D point $\mathbf{S}_p$ a point cloud $\mathbf{R}_p$ that consists of feature points lying within a certain distance from $\mathbf{S}_p$. First, the directions of maximum and minimum curvatures are computed for each $\mathbf{R}_p$ using local quadratic surface fit. For a target whose curve lies along one direction, as assumed in this paper, the minimum principal curvature must be 0, and its direction must be the same for all the feature points. In practice, however, there exists some fluctuation in the directions of minimum curvature, due to the estimation errors. Thus, a voting method is applied to eliminate outliers and the dominant direction $\mathbf{V}_{\min} = \left(v_{mx}, v_{my}, v_{mz}\right)^T$ of minimum principal curvatures for the whole target is determined.

Next, 3D position $\mathbf{S}_p$ for each feature point is projected to a plane whose normal vector $\mathbf{N}$ coincides with $\mathbf{V}_{\min}$ i.e. $P(x, y, z) = v_{mx}x + v_{my}y + v_{mz}z = 0$. The projected 2D coordinates $\left(\bar{x}_p, \bar{y}_p\right)$ of $\mathbf{S}_p$ is given as follows:

$$\begin{pmatrix} \bar{x}_p \\ \bar{y}_p \end{pmatrix} = \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \mathbf{S}_p \, , \tag{4}$$

where $\mathbf{V}_1$ is a unit vector parallel to the principle axis of inertia of the projected 2D coordinates $\left(\bar{x}, \bar{y}\right)$, and $\mathbf{V}_2$ is a unit vector that is perpendicular to $\mathbf{V}_1$ and $\mathbf{V}_{\min}$, i.e. $\mathbf{V}_2 = \mathbf{V}_1 \times \mathbf{N}$.

Finally, the target shape parameter $\left(a_1, \ldots, a_m\right)$ is obtained by fitting the following variable-order polynomial equation to the projected 2D coordinates $\left(\bar{x}, \bar{y}\right)$.

$$\bar{y} = f\left(\bar{x}\right) = \sum_{i=1}^{m} a_i \bar{x}^i \, . \tag{5}$$

Here, the optimal order m in the above equation is automatically determined by using geometric AIC [17].

In the case where the target is composed of multiple curved surfaces, e.g. the thick bound book shown in Figure 1(a), the target is first divided with a line where the normal vector of the locally fitted quadratic surface varies discontinuously, and then the shape parameter is computed for each part of the target. The estimated shape is used for generating a dewarped mosaic image in the next process, as well as for removing the perspective distortion in the reappearing feature detection process (Figure 2(c)).

Step (f): Mosaic Image Generation.

Finally, a mosaic image is generated using extrinsic camera parameters and surface shape parameters. The super-resolution and shading correction are implemented here regarding coordinate transformation below. Let us consider a 2D coordinate (m,n) on the dewarped mosaic image, as shown in Figure 6. The high-definite pixel value at (m,n) on the dewarped mosaic image is estimated from the intrinsic pixel values at all the corresponding coordinates (u_f, v_f) in the input subimage [11,12].

The relation between (m,n) and its corresponding 3D coordinate $(\bar{x}, f(\bar{x}), \bar{z})$ on the fitting surface is given as follows:

$$(m,n) = \left(\int_0^{\bar{x}} \sqrt{1+\left(\frac{d}{dx}f(x)\right)^2}\,dx, \bar{z} \right). \tag{6}$$

The coordinate $(\bar{x}_p, f(\bar{x}_p), \bar{z})$ is transferred to the corresponding 2D coordinate (u_f, v_f) on the *f*-th image plane by the following equation.

$$\begin{pmatrix} au_f \\ av_f \\ a \end{pmatrix} = \mathbf{M}_f \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \mathbf{N}^T \end{pmatrix} \begin{pmatrix} \bar{x} \\ f(\bar{x}) \\ \bar{z} \end{pmatrix}. \tag{7}$$

Assumption of the flat target shape (flat mode) simplifies these as planar perspective transformations.

3. Experiments

We have developed a video mosaicing prototype system that consists of a mobile PC (Pentium-M 1. 2GHz, Memory 1GB) and a USB web-cam. The appearance of the prototype system has been shown in Figure 3. In our system, a user starts image capturing, referring the monitor of mobile computer. The real-time stage for image capturing is automatically finished when a video buffer becomes full. Experiments have been carried out for flat and curved documents. In these experiments, the intrinsic camera parameters are calibrated in advance by Tsai's method [18], and are fixed throughout image capturing. Note that in the current version of the prototype system, the curved mode is not implemented on the mobile system. Thus, the latter experiment for the curved surface is carried out by using a desktop PC (Xeon 3.2GHz, Memory 2GB), and the initial camera parameter estimation is processed offline after image capturing using an IEEE 1394 web-cam. Experimental results chiefly focused on 3D reconstruction are shown below.

3.1. Flat target

As shown in Figure 7, a flat target document is captured as 150 frame images of 640x480 pixels at 6 fps with initial camera parameter estimation. Image features tracked in the real-time stage are depicted with cross marks. Note that none of the input image planes are parallel to the target document. Figure 8 illustrates estimated extrinsic camera parameters and feature positions on the mosaic image plane. The curved line shows the estimated camera path and pyramids show the camera postures every 10 frames. The generated mosaic image is shown in Figure 9. We can confirm that the slanted effect is correctly removed in the final mosaic image by comparing the result with the homography-based result shown in Figure 1(b) where the same input images and the same feature trajectories with this experiment were used. Although the target paper is not perfectly flat, there are no apparent distortions in the final mosaic image. The performance of the

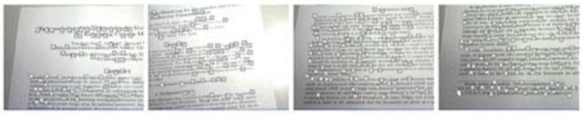
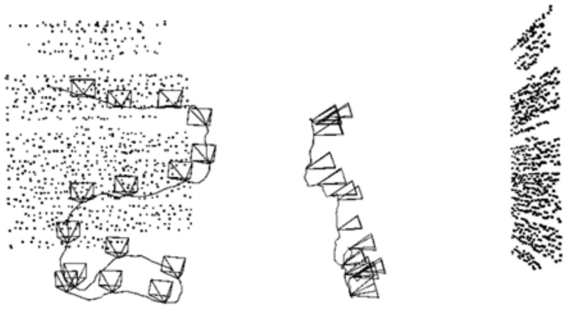


Figure 7. Input subimage samples and tracked features for flat target.



(a) Top view (b) Side view

Figure 8. Estimated extrinsic camera parameter and 3D feature distribution for flat document.

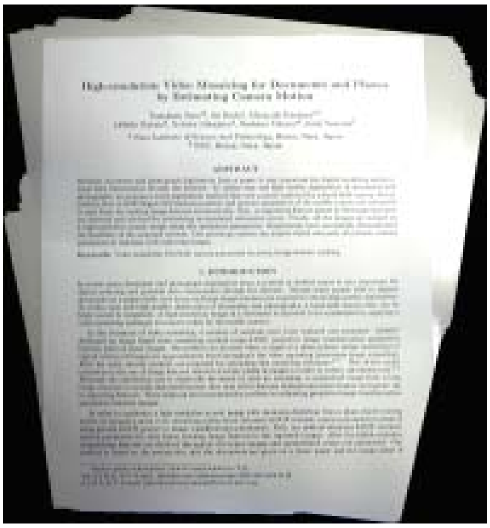


Figure 9. Reconstructed mosaic image.

prototype system for this sequence is as follows: 6 fps for image acquisition and initial parameter estimation with preview of mosaicing, 17 seconds for camera parameter refinement, and 39 seconds for generating the final mosaic image.

3.2. Curved target

An example of curved documents used in experiments is shown in Figure 10. The target is composed of 2 pages of curved surfaces: one page with texts and the other with pictures and figures. The target is captured with a web-cam as a video of 200 frames at 7.5 fps, and is used as an input to our system. Sampled frames of the captured video are shown in Figure 11. Tracked feature points are depicted with cross marks. The 3D reconstruction result obtained by the proposed method is shown in Figure 12. The curved line shows the camera path, pyramids show the camera postures every 10 frames, and the point cloud shows 3D positions of feature points. As can be seen, the point cloud coincides with the shape of the thick bound book. The shape of the target is estimated after 3 time iterations of reappearing feature detection and the surface fitting process. The estimated shape of the target is shown in Figure 13. In the proposed method, the optimal order of the polynomial surface fitted to the target is automatically determined by geometric AIC. In this experiment, the order is 5 and 4 for the left and right pages, respectively. The dewarped mosaic images before and after shadow removal are shown in Figure 14(a) and (b), respectively. As can be seen, the distortion on the target has been removed in the resultant image. The performance of the system on the desktop PC for this sequence is as follows: 27 seconds for initial 3D reconstruction, 71 seconds for camera parameter refinement, and 113 seconds for generating the final mosaic image. Some other mosaicing results for curved targets are shown in Figure 15.

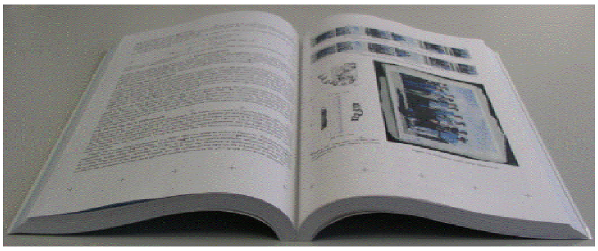
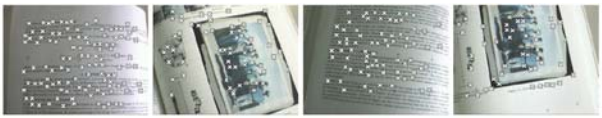
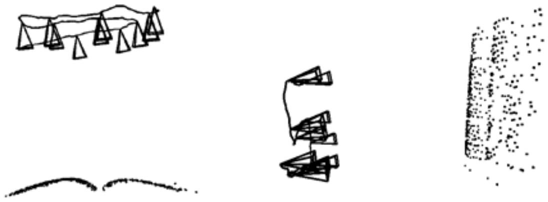


Figure 10. Curved target.



1st frame 66th frame 123rd frame 200th frame

Figure 11. Input subimage samples for curved target.



(a) Front view (b) Side view

Figure 12. Estimated extrinsic camera parameter and 3D feature distribution for curved document.

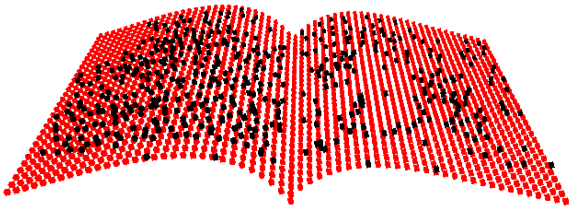
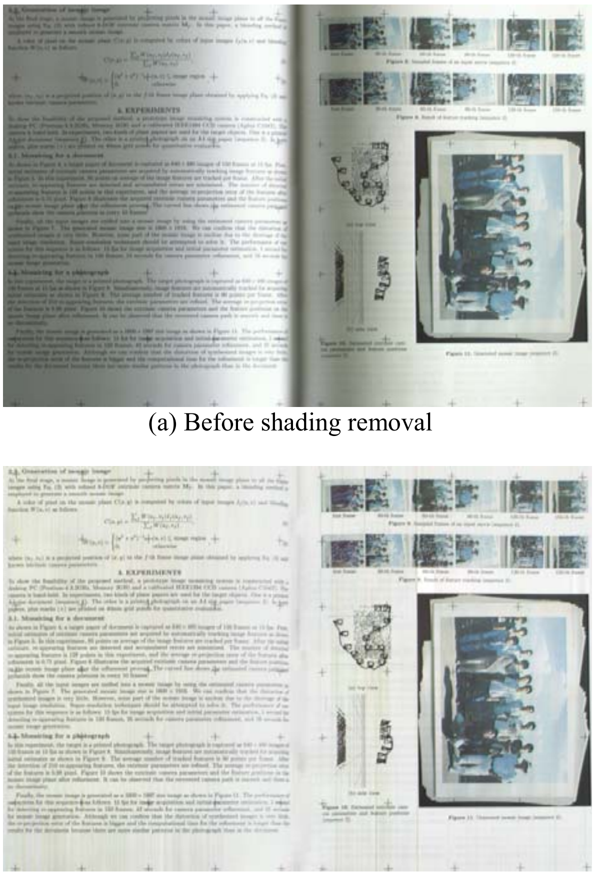


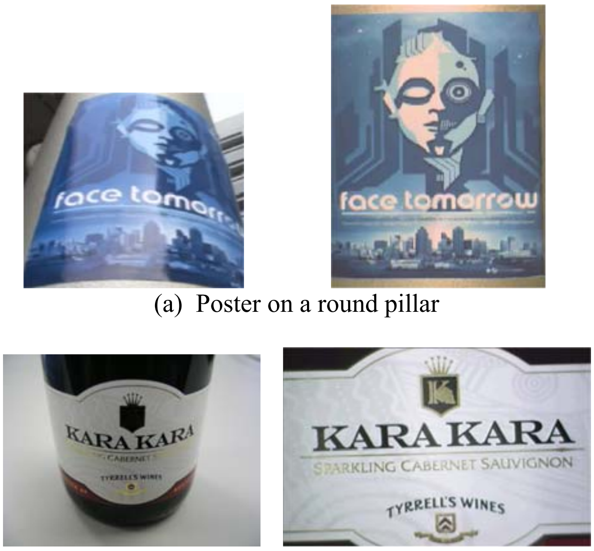
Figure 13. Estimated shape.



(a) Before shading removal

(b) After shading removal

Figure 14. Reconstructed mosaic image.



(a) Poster on a round pillar

(b) Bottle label

Figure 15. Mosaicing result examples. Left: target appearances. Right: reconstructed images.

4. Conclusion

A novel video mosaicing method for generating a high-resolution and distortion-free mosaic image for flat and curved documents has been proposed. With this method based on 3D reconstruction, the 6-DOF camera motion and the shape of the target document are estimated without any extra devices. In experiments, a prototype system of mobile video mosaicing has been developed and has been successfully demonstrated. For a flat target, a mosaic image without the slanted effect is successfully generated. Although the curved mode has not yet been implemented on the current version of mobile system, we have shown the offline result for curved documents generated by desktop PC. In the curved mode, assuming the curve of the target lies along one direction, the shape model is fitted to the feature point cloud and a dewarped image without shadow is automatically generated. Our future work involves reducing the computational cost in the curved mode and improving the accuracy of 3D reconstruction around the inner margin of a book.

References

[1] K. Junga, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: a survey," *Pattern Recognition*, Vol. 37, No.5, 2004, pp. 977 – 997.

[2] A. Rosenfeld, D. Doermann, and D. DeMenthon, *Video Mining*, Kluwer, Massachusetts, 2003.

[3] D. Doermann, J. Liang, and H. Li, "Progress in camera-based document image analysis," *Proc. Int. Conf. Document Analysis and Recognition*, 2003, pp. 606–616.

[4] R. Szeliski, "Image Mosaicing for Tele-Reality Applications," *Proc. IEEE Workshop Applications of Computer Vision*, pp. 230-236, 1994.

[5] S. Takeuchi, D. Shibuichi, N. Terashima, and H. Tominage, "Adaptive Resolution Image Acquisition Using Image Mosaicing Technique from Video Sequence," *Proc. Int. Conf. Image Processing*, vol. I, pp. 220-223, 2000.

[6] C. T. Hsu, T. H. Cheng, R. A. Beuker, and J. K. Hong, "Feature-based Video Mosaicing," *Proc. Int. Conf. Image Processing*, Vol. 11, pp. 887-890, 2000.

[7] M. Lhuillier, L. Quan, H. Shum, and H. T. Tsui, "Relief Mosaicing by Joint View Triangulation," *Proc. Int. Conf. Computer Vision and Pattern Recognition*, vol. 19, pp. 785-790, 2001.

[8] P. T. McLauchlan, and A. Jaenicke, "Image Mosaicing Using Bundle Adjustment," *Image and Vision Computing*, 20, pp. 751-759, 2002.

[9] D. W. Kim and K. S. Hong, "Fast Global Registration for Image Mosaicing," *Proc. Int. Conf. Image Processing*, 11, pp. 295-298, 2003.

[10] P. Grattono, and M. Spertino, "A Mosaicing Approach for the Acquisition and Representation of 3D Painted Surfaces for Conservation and Restoration Purposes," *Machine Vision and Applications*, Vol. 15, No. 1, pp. 1-10, 2003.

[11] D. Capel, *Image Mosaicing and Super-resolution*, Springer-Verlag, London, 2004.

[12] Y. Weiss, "Deriving intrinsic images from image sequences," *Proc. Int. Conf. Computer Vision*, pp. 68-75, 2001.

[13] T. Sato, M. Kanbara, N. Yokoya, and H. Takemura, "Dense 3D Reconstruction of an Outdoor Scene by Hundreds-baseline Stereo Using a Hand-held Video Camera," *Int. J. Computer Vision*, Vol. 47, No. 1-3, pp. 119-129, 2002.

[14] C. Harris, and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Alvey Vision Conf.*, pp. 147-151, 1988.

[15] MA Fischler, and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, Vol. 24, No. 6, pp. 381-395, 1981.

[16] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle Adjustment - a Modern Synthesis," *Proc. Int. Workshop Vision Algorithms*, pp. 298-372, 1999.

[17] K. Kanatani, "Geometric Information Criterion for Model Selection,", *Int. J. Computer Vision*, Vol. 26, No. 3, pp. 171-189, 1998.

[18] R. Y. Tsai, "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 364-374, 1986.

Contest

Document Image Dewarping Contest

Faisal Shafait
German Research Center for Artificial Intelligence (DFKI),
Kaiserslautern, Germany
faisal.shafait@dfki.de

Thomas M. Breuel
Department of Computer Science
Technical University of Kaiserslautern, Germany
tmb@informatik.uni-kl.de

Abstract

Dewarping of documents captured with hand-held cameras in an uncontrolled environment has triggered a lot of interest in the scientific community over the last few years and many approaches have been proposed. However, there has been no comparative evaluation of different dewarping techniques so far. In an attempt to fill this gap, we have organized a page dewarping contest along with CBDAR 2007. We have created a dataset of 102 documents captured with a hand-held camera and have made it freely available online. We have prepared text-line, text-zone, and ASCII text ground-truth for the documents in this dataset. Three groups participated in the contest with their methods. In this paper we present an overview of the approaches that the participants used, the evaluation measure, and the dataset used in the contest. We report the performance of all participating methods. The evaluation shows that none of the participating methods was statistically significantly better than any other participating method.

1 Introduction

Research on document analysis and recognition has traditionally been focused on analyzing scanned documents. Many novel approaches have been proposed over the years for performing page segmentation [1] and optical character recognition (OCR) [2] on scanned documents. With the advent of digital cameras, the traditional way of capturing documents is changing from flat-bed scans to capture by hand-held cameras [3, 4]. Recognition of documents captured with hand-held cameras poses many additional technical challenges like perspective distortion, non-planar surfaces, low resolution, uneven lighting, and wide-angle-lens distortions [5]. One of the main research directions in camera-captured document analysis is to deal with the page curl and perspective distortions. Current document analysis and optical character recognition systems do not expect these types of artifacts, and show poor performance when applied directly to camera-captured documents. The goal of page dewarping is to flatten a camera captured document such that it becomes readable by current OCR systems.

Over the last decade, many different approaches have been proposed for document image dewarping [5]. These approaches can be grouped into two broad categories according to the acquisition of images:

1. 3-D shape reconstruction of the page using specialized hardware like stereo-cameras [6, 7], structured light sources [8], or laser scanners [9].

2. reconstruction of the page using a single camera in an uncontrolled environment [10, 11, 12]

The first approaches proposed in the literature for page dewarping were those based on 3-D shape reconstruction. One of the major drawbacks of the approaches requiring specialized hardware is that they limit the flexibility of capturing documents with cameras, which is one of the most important features of camera-based document capture. Therefore, the approaches based on a single camera in an uncontrolled environment have caught more attention recently. The approach in [12] claims to be the first dewarping approach for documents captured with hand-held cameras in an uncontrolled environment. It is interesting to note that the approaches in [10, 11, 12], which were all published in 2005, actually served as a trigger for research in analyzing documents captured with a hand-held camera and many other approaches like [13, 14, 15] have emerged in the following years. Despite the existence of so many approaches for page dewarping, there is no comparative evaluation so far. One of the main problems is that the authors use their own datasets for evaluation of their approaches, and these datasets are not available to other researchers.

As a first step towards comparative evaluation of page dewarping techniques, we have organized a page dewarping contest along with CBDAR 2007. For this purpose we have developed a dataset of camera captured documents and have prepared ground-truth information for text-lines, text-zone, and ASCII text for all documents in the dataset (Section 2). Three groups participated in the contest. The dataset was given to the participants, and they were given a time frame of two weeks to return flattened document images, along with a brief summary of their methods. The description of the participating methods is given in Section 3. The documents returned by the participants were processed by an OCR system to compare and evaluate their performance. The results of the participating methods are discussed in Section 4 followed by a conclusion in Section 5.

2 DFKI-1 Warped Documents Dataset

To compare different dewarping approaches on a common dataset, we have prepared a ground-truthed database of camera captured documents. The dataset contains 102 binarized images of pages from several technical books captured by an off-the-shelf digital camera in a normal office environment. No specialized hardware or lighting was used. The captured documents were binarized using a local adaptive thresholding technique [11]. Some sample documents from the dataset are shown in Figure 8.

The following types of ground-truth are provided with the dataset:

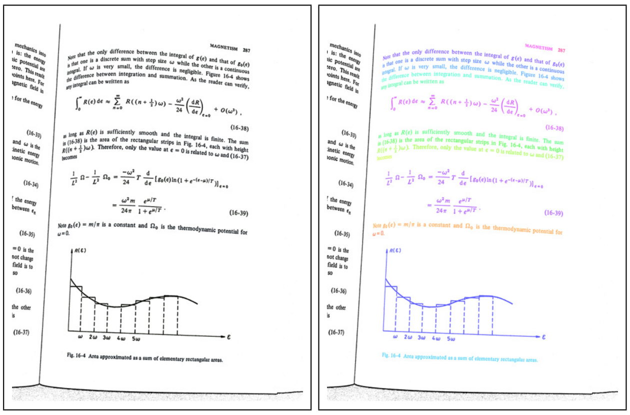
1. ground-truth text-lines in color-coded format (Fig 1)

2. ground-truth zones in color-coded format (Fig 1)

3. ground-truth ASCII text in plain text format

Many approaches for dewarping use detection of curved text-lines as a first step [11, 15]. The purpose of providing text-line and text-zone level ground-truth is to assist the researchers in quantitatively measuring the performance of this important intermediate step. ASCII text ground-truth is intended for use as the overall performance measure of a dewarping system by using OCR on the dewarped document. The dataset is publicly available for download from *http://www.iupr.org/downloads/data*.

The dataset is not split into training and test set, because some algorithms need larger training sets as compared to others. It is expected that when other researchers use this dataset, they will split it into test and training sets as per requirements.



MAGNETISM 287

Note that the only difference between the integral of $g(\epsilon)$ and that of $g_0(\epsilon)$ is that one is a discrete sum with step size ω while the other is a continuous integral. If ω is very small, the difference is negligible. Figure 16-4 shows the difference between integration and summation. As the reader can verify, any integral can be written as

$$\int_0^\infty R(\epsilon)\,d\epsilon \approx \sum_{n=0}^{\infty} R((n+\tfrac{1}{2})\omega) - \frac{\omega^2}{24}\left(\frac{dR}{d\epsilon}\right)_{\epsilon=0} + O(\omega^3)\,, \quad (16\text{-}38)$$

as long as $R(\epsilon)$ is sufficiently smooth and the integral is finite. The sum in (16-38) is the area of the rectangular strips in Fig. 16-4, each with height $R((n+\frac{1}{2})\omega)$. Therefore, only the value at $\epsilon = 0$ is related to ω and (16-37) becomes

$$\frac{1}{L^2}\,\Omega - \frac{1}{L^2}\,\Omega_0 = \frac{-\omega^2}{24}\,T\,\frac{d}{d\epsilon}\left[g_0(\epsilon)\ln(1+e^{-(\epsilon-\mu)/T})\right]_{\epsilon=0}$$

$$= \frac{\omega^2 m}{24\pi}\,\frac{e^{\mu/T}}{1+e^{\mu/T}}\,. \quad (16\text{-}39)$$

Note $g_0(\epsilon) = m/\pi$ is a constant and Ω_0 is the thermodynamic potential for $\omega = 0$.

Fig. 16-4 Area approximated as a sum of elementary rectangular areas.

Figure 1. An example image (left) showing the ground-truth text-line and text-zone level information. The green channel contains the label of document zone type (text, graphics, math, table, image), the red channel contains the paragraph information for text-zones in reading order, and the blue channel contains the text-line information. For more information on pixel-accurate representation of page segmentation, please refer to [16]. The right image just replaces all the colors in the original ground-truth image with different visually distinguishable colors for visualization purposes.

3 Participating Methods

Three methods for document image dewarping were presented for participation in the contest by different research groups:

1. Continuous skeletal image representation for document image dewarping[1]

2. Segmentation based document image dewarping[2]

3. Coordinate transform model and document rectification for book dewarping[3]

[1] A. Masalovitch, L. Mestetskiy. Moscow State University, Moscow, Russia. anton_m@abbyy.com, l.mest@ru.net

[2] B. Gatos, N. Stamatopoulos, K. Ntirogiannis and I. Pratikakis. Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", GR-153 10 Agia Paraskevi, Athens, Greece. http://www.iit.demokritos.gr/∼bgat/, {bgat,nstam,ipratika}@iit.demokritos.gr

[3] W. Li, B. Fu, M. Wu. Department of Computer Science and Technology, Peking University, Beijing 100871, China. {lwx,fubinpku,wuminghui}@pku.edu.cn

The text in the next sub-sections summarizes these methods and is based on the description of the methods provided by the participants.

3.1 Continuous skeletal image representation for document image dewarping (SKEL) [17]

This approach for image dewarping is based on the construction of outer skeletons of text images. The main idea of this algorithm is based on the fact that it is easy to mark up long continuous branches that define inter-linear spaces of the document in outer skeletons. Such branches can be approximated by cubic Bezier curves to find a specific deformation model of each inter-linear space of the document. On the basis of a set of such inter-linear space approximations, the whole approximation of the document is built in the form of a two-dimensional cubic Bezier patch. Then, the image can be dewarped using the obtained approximation of the image deformation.

3.1.1 Problem definition

Consider an image $I(x, y)$, where I is the color of the image pixel with coordinates (x, y). The goal of page dewarping is to develop a continuous vector function $D(x, y)$ to obtain a dewarped image in the form: $\overline{I}(x, y) = I(D_x(x, y), D_y(x, y))$. This function will be the approximation of the whole image deformation.

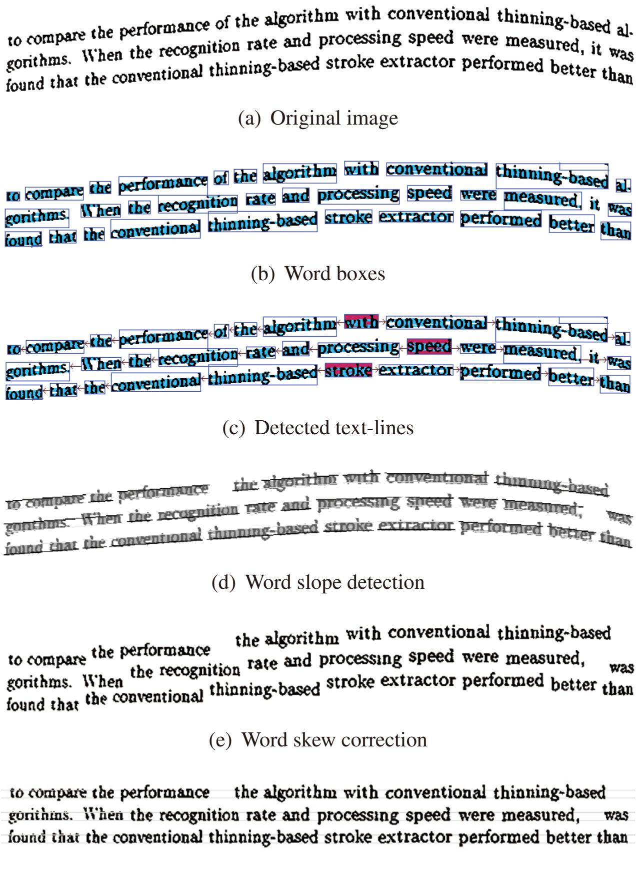
3.1.2 Main idea of the algorithm

The main idea of this algorithm is that in an outer skeleton of a text document image, one can easily find branches that lie between adjacent text-lines. Then, one can use this separation branches to approximate deformation of inter-linear spaces on the image. The proposed algorithm consists of the following steps:

1. A continuous skeletal representation of an image is built. The skeleton of an area is a set of points, such that for each point there exist no less than two nearest points on the border of the area. As border representation, polygons of minimal perimeter that enclose black objects on a picture are used. Methods exist that allow building of a continuous skeleton in time $O(n \log(n))$ [18].

2. The skeleton is filtered (useless bones are deleted).

3. All branches of the skeleton are clustered by their length and angle to find out horizontal and vertical branches.



(a) Original image

(b) Word boxes

(c) Detected text-lines

(d) Word slope detection

(e) Word skew correction

(f) Dewarped document

Figure 2. Example of the intermediate steps of page deskewing with the SEG approach.

4. The list of horizontal branches is filtered to leave only branches that lie between different text-lines.

5. A cubic Bezier approximation is built for each branch.

6. A two-dimensional Bezier patch is built that approximates all obtained curves. The patch is represented in the following form:

$$D(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} P_{ij} b_{i,3}(x) b_{j,3}(y) \qquad (1)$$

   where $b_{r,3}(t)$ is a cubic Bernstein polynomial.

The patch thus obtained approximates the deformation function of the whole page.

3.2 Segmentation based document image dewarping (SEG) [19]

This technique enhances the quality of documents captured by a digital camera relying upon

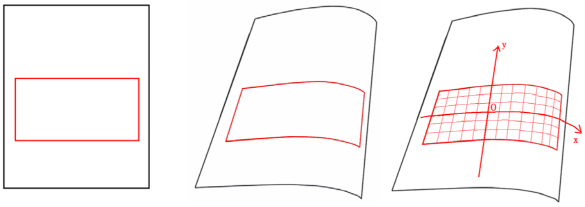


Figure 3. An example of image distortion of a flat area on a page when captured by a hand-held camera. The right-most image shows the curved coordinate net used in the CTM method.

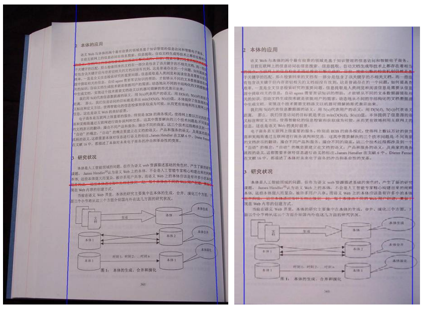


Figure 4. Illustration of document image before and after rectification with the CTM method.

1. automatically detecting and cutting out noisy black borders as well as noisy text regions appearing from neighboring pages
2. text-lines and words detection using a novel segmentation technique appropriate for warped documents
3. a first draft binary image dewarping based on word rotation and translation according to upper and lower word baselines
4. a recovery of the original warped image guided by the draft binary image dewarping result

In this approach, black border as well as neighboring page detection and removal is done followed by an efficient document image dewarping based on text-line and word segmentation [19]. The methodology for black border removal is mainly based on horizontal and vertical profiles. First, the image is smoothed, then the starting and ending offsets of borders and text regions are calculated. Black borders are removed by also using the connected components of the image. We detect noisy text regions appearing from neighboring page with the help of the signal cross-correlation function.

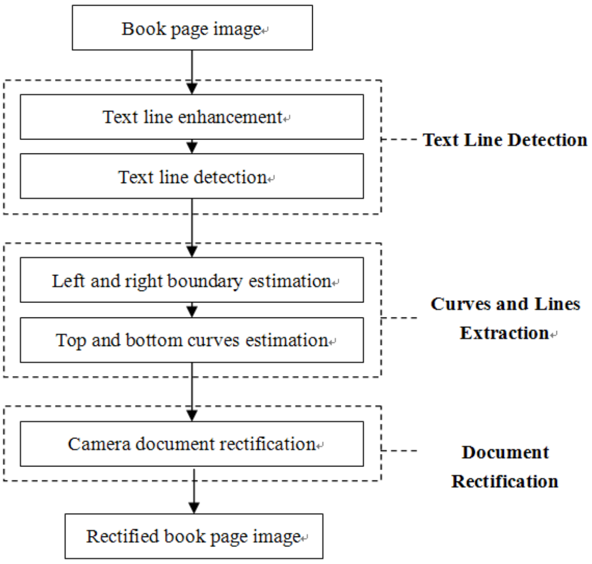

Book page image
Text line enhancement
Text line detection
Text Line Detection
Left and right boundary estimation
Top and bottom curves estimation
Curves and Lines Extraction
Camera document rectification
Document Rectification
Rectified book page image

Figure 5. A flowchart of the CTM method.

At a next step, all words are detected using a proper image smoothing (Figure 2(b)). Then, horizontally neighboring words are consecutively linked in order to define text-lines. This is accomplished by consecutively extracting right and left neighboring words to the first word detected after top-down scanning (Figure 2(c)). For every detected word, the lower and upper baselines are calculated, which delimit the main body of the word, based on a linear regression which is applied on the set of points that are the upper or lower black pixels for each word image column [20]. The slope of each word is derived from the corresponding baselines slopes (Figure 2(d)). All detected words are then rotated and shifted (Figure 2(e)) in order to obtain a first draft estimation of the binary dewarped image. Finally, a complete restoration of the original warped image is done guided by the draft binary dewarping result of the previous stage. Since the transformation factors for every pixel in the draft binary dewarped image have been already stored, the reverse procedure is applied on the original image pixels in order to retrieve the final dewarped image. For all pixels for which transformation factors have not been allocated, the transformation factors of the nearest pixel are used.

3.3 Coordinate transform model and document rectification for book dewarping (CTM) [21]

This method uses a coordinate transform model and document rectification process for book dewarping. This model assumes that the book surface is a cylinder. It can handle both perspective distortion and book surface warping

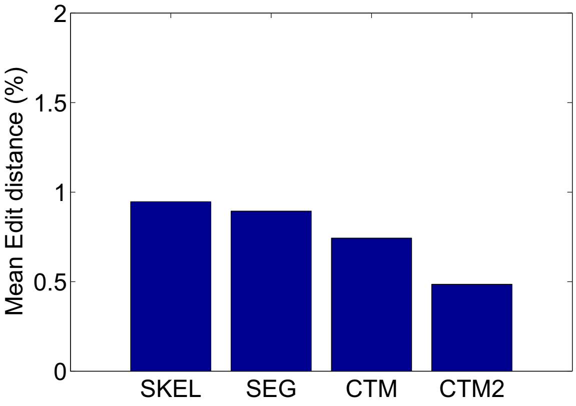


Figure 6. Mean edit distance of the text extracted by running Omnipage on the dewarped documents. Note that CTM2 just adds to CTM some post-processing steps to remove graphics and images from the dewarped documents.

problems. The goal is to generate a transformation to flatten the document image to its original shape (see Figure 3). The transformation is a mapping from the curved coordinate system to a Cartesian coordinate system. Once a curved coordinate net is set up on the distorted image as shown in Figure 3, the transformation can be done in two steps: First, the curved net is stretched to a straight one, and then adjusted to a well-proportioned square net.

According to the transform model, two line segments and two curves are needed to dewarp a cylinder image. Therefore, the left and right boundaries and top and bottom curves in book images are found for the rectification as shown in Figure 4.

The rectification process involves three steps: 1) the text-line detection, 2) left and right boundary estimation and top and bottom curves extraction, and 3) document rectification. The flowchart of the rectification process is illustrated in Figure 5.

As an additional post-processing step, the participants used their programs to remove graphics and images from the processed pages. The results thus produced are referred to as **CTM2**.

4 Experiments and Results

The results of the participating methods on some example documents from the dataset are shown in Figure 8. The dewarped documents returned by the participants were processed through Omnipage Pro 14.0, a commercial OCR system. After obtaining the text from the OCR software, the edit distance with the ASCII text ground-truth was used as the error measure. Although OCR accuracy is a good measure for the performance of dewarping on text regions, it does not measure how well the dewarping algorithm worked on the non-text parts, like math or graphics regions. Despite this limitation, we used the OCR accuracy because it is the most widely-used measure for measuring performance of dewarping systems [5].

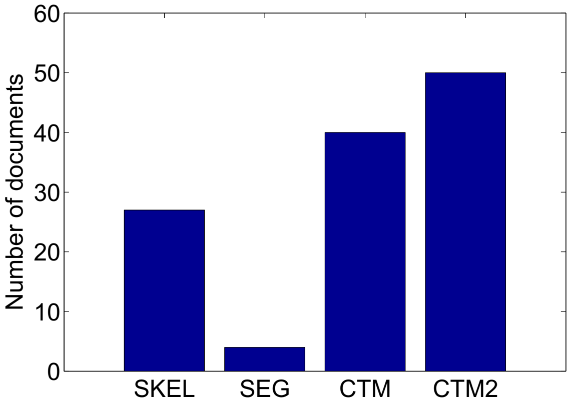
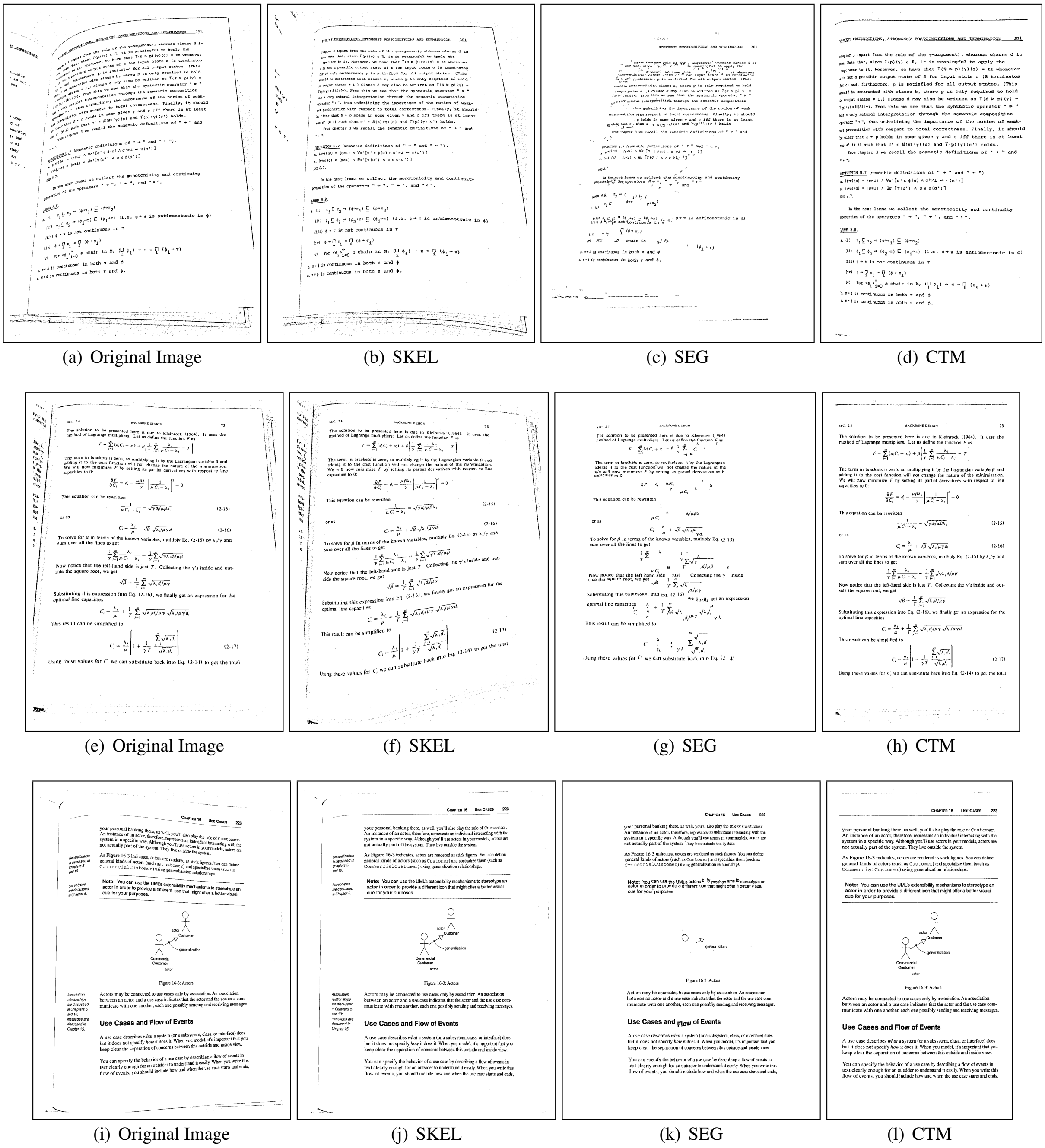


Figure 7. Number of documents for each algorithm on which it had the lowest edit distance among the participating methods.

The mean edit distance of the participating methods is shown in Figure 6. The graph shows that the CTM technique performs best on the test data, and its results further improve after post-processing to remove graphics and images. This is because the ground-truth ASCII text contains text coming only from the textual parts of the documents, so the text that is present in graphics or images is ignored. Hence, the dewarped documents that contain text inside graphics regions get higher edit distances.

To analyze whether one algorithm is uniformly better than the other algorithms, we plotted the number of documents for each algorithm on which it had the lowest edit distance on character basis (Figure 7). If there was a tie between more than one methods for the lowest error rate on a particular document, all algorithms were scored for that document. Interestingly, the results show that the SEG method achieves the lowest error rate in only four documents. Here again the CTM2 method proves to be the best for the highest number of documents.

The analysis of the difference in the performance of the participating algorithms was done using a box plot (Figure 9). The boxes in the box plot represent the interquartile range, i.e. they contain the middle 50% of the data. The lower and upper edges represent the first and third quartiles, whereas the middle line represents the median of the data.



(a) Original Image (b) SKEL (c) SEG (d) CTM

(e) Original Image (f) SKEL (g) SEG (h) CTM

(i) Original Image (j) SKEL (k) SEG (l) CTM

Figure 8. Example results of the participants. For image 8(a), the SKEL and SEG methods remove page curl distortion, but could not handle perspective distortion. In image 8(e), the SKEL method was misled by the formulas and did not dewarp it correctly. In image 8(i), the SEG and CTM methods removed some text parts that were present near the left border of the page.

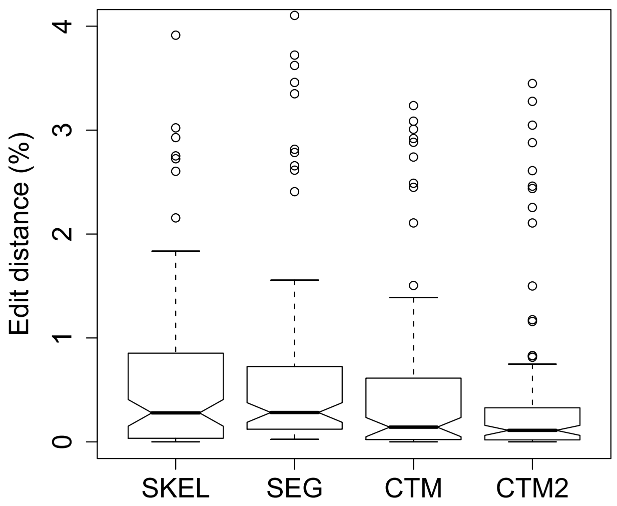


Figure 9. A box plot of the percentage edit distance for each algorithms. Overlapping notches of the boxes show that none of the participating algorithms is statistically significantly better than any other algorithms.

The notches represent the expected range of the median. The 'whiskers' on the two sides show inliers, i.e. points within 1.5 times the interquartile range. The outliers are represented by small circles outside the whiskers. Figure 9 shows that the expected range of medians of the edit distance overlaps for all the algorithms. Hence, it can be concluded that none of the participating algorithms is statistically significantly better than any other algorithm.

5 Conclusion

The purpose of the dewarping contest was to take a first step towards a comparative evaluation of dewarping techniques. Three groups participated in the competition with their methods. The results showed that the coordinate transform model (CTM) presented by Wenxin Li et al. performed better than the other two methods, but the difference was not statistically significant. Overall, all participating methods worked well and the mean edit distance was less than 1% for each of them. We have made the dataset used in the contest publicly available so that other researchers can use the dataset to evaluate their methods.

Acknowledgments

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project IPeT (01 IW D03).

References

[1] F. Shafait, D. Keysers, and T.M. Breuel. Performance comparison of six algorithms for page segmentation. In *7th IAPR Workshop on Document Analysis Systems*, pages 368–379, Nelson, New Zealand, Feb. 2006.

[2] S. Mori, C.Y. Suen, and K. Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7):1029–1058, 1992.

[3] M. J. Taylor, A. Zappala, W. M. Newman, and C. R. Dance. Documents through cameras. In *Image and Vision Computing 17*, volume 11, pages 831–844, September 1999.

[4] T.M. Breuel. The future of document imaging in the era of electronic documents. In *Int. Workshop on Document Analysis*, Kolkata, India, Mar. 2005.

[5] J. Liang, D. Doermann, and H. Li. Camera-based analysis of text and documents: a survey. *Int. Jour. of Document Analysis and Recognition*, 7(2-3):84–104, 2005.

[6] A. Ulges, C. Lampert, and T. M. Breuel. Document capture using stereo vision. In *Proceedings of the ACM Symposium on Document Engineering*, pages 198–200. ACM, 2004.

[7] A. Yamashita, A. Kawarago, T. Kaneko, and K.T. Miura. Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. In *Proceedings of 17th International Conference on Pattern Recognition (ICPR2004), Vol.1*, pages 482–485, 2004.

[8] M.S. Brown and W.B. Seales. Document restoration using 3d shape: A general deskewing algorithm for arbitrarily warped documents. In *International Conference on Computer Vision (ICCV01)*, volume 2, pages 367–374, July 2001.

[9] M. Pilu. Deskewing perspectively distorted documents: An approach based on perceptual organization. *HP White Paper*, May 2001.

[10] L. Zhang and C.L. Tan. Warped image restoration with applications to digital libraries. In *Proc. Eighth Int. Conf. on Document Analysis and Recognition*, pages 192–196, Aug. 2005.

[11] A. Ulges, C.H. Lampert, and T.M. Breuel. Document image dewarping using robust estimation of curled text lines. In *Proc. Eighth Int. Conf. on Document Analysis and Recognition*, pages 1001–1005, Aug. 2005.

[12] J. Liang, D.F. DeMenthon, and D. Doermann. Flattening curved documents in images. In *Proc. Computer Vision and Pattern Recognition*, pages 338–345, June 2005.

[13] S. Lu and C.L. Tan. The restoration of camera documents through image segmentation. In *7th IAPR Workshop on Document Analysis Systems*, pages 484–495, Nelson, New Zealand, Feb. 2006.

[14] S. Lu and C.L. Tan. Document flattening through grid modeling and regularization. In *Proc. 18th Int. Conf. on Pattern Recognition*, pages 971–974, Aug. 2006.

[15] B. Gatos and K. Ntirogiannis. Restoration of arbitrarily warped document images based on text line and word detection. In *Fourth IASTED Int. Conf. on Signal Processing, Pattern Recognition, and Applications*, pages 203–208, Feb. 2007.

[16] F. Shafait, D. Keysers, and T.M. Breuel. Pixel-accurate representation and evaluation of page segmentation in document images. In *18th Int. Conf. on Pattern Recognition*, pages 872–875, Hong Kong, China, Aug. 2006.

[17] A. Masalovitch and L. Mestetskiy. Usage of continuous skeletal image representation for document images de-warping. In *2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, Sep. 2007. Accepted for publication.

[18] L.M. Mestetskiy. Skeleton of multiply connected polygonal figure. In *Proc. 15th Int. Conf. on Computer Graphics and Applications*, Novosibirsk, Russia, June 2005.

[19] B. Gatos, I. Pratikakis, and K. Ntirogiannis. Segmentation based recovery of arbitrarily warped document images. In *Proc. Int. Conf. on Document Analysis and Recognition*, Curitiba, Brazil, Sep. 2007. Accepted for publication.

[20] U.V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. Jour. of Pattern Recognition and Artifical Intelligence*, 15(1):65–90, 2001.

[21] B. Fu, M. Wu, R. Li, W. Li, and Z. Xu. A model-based book dewarping method using text line detection. In *2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, Sep. 2007. Accepted for publication.

Author Index