社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

レイアウトの変動にも対応できる文書画像検索法

上田　敬介†　　黄瀬　浩一†

† 大阪府立大学大学院工学研究科　〒 599–8531 大阪府堺市中区学園町 1–1
E-mail:　†ueda@m.cs.osakafu-u.ac.jp, †kise@cs.osakafu-u.ac.jp

あらまし　レイアウトが変更されていても，コンテンツが一致すれば検索が可能な文書画像検索法を提案する．文書画像検索の既存手法は大きく分けて 2 種類ある．1 つは，各文字や各単語の近傍位置関係を特徴として検索する手法である．もう 1 つは，文字認識を行い，文書をコード化して検索を行う手法である．しかし，これらの手法にはそれぞれ問題点がある．前者の手法では，データベースの文書画像とレイアウトが違う文書画像を与えると検索ができなくなる．後者の手法では，レイアウト変動には柔軟であるが，処理時間が長いという問題点がある．そこで本研究では，レイアウトの変動にも対応でき，文字認識ほど厳密な処理を行わない手法を提案する．レイアウトの違う文書 300 ページをクエリ画像とし，データベースの画像 10,000 枚に対して検索実験を行ったところ，認識精度 93.7%，検索時間 417[ms] を得た．これは OCR を用いて得られる認識精度 99.0%には劣るものの，検索時間は 1/4 となっていることから高速な検索が可能であることがわかった．

キーワード　文書画像検索, カメラベース, OCR, k-NN

1. はじめに

デジタルカメラの高機能化，一般化に伴い，その情報デバイスとしての利用が注目されている．最も素朴な利用法の一つは検索である．すなわち，デジタルカメラで撮影した対象物の画像を検索質問として，その対象物に関する情報を検索することである．

文書画像検索とは，手元にある未知の文書を撮影し，その撮影画像を検索質問としてデータベースから同一文書を検索する処理である．現在，電子書籍の発達に伴い，Layered Reading [1] という情報サービスが注目されている．これは電子書籍の上に新しいレイヤを設け，付加情報を重ねて表示するというものである．文書画像検索が高速かつ容易になれば，現在は電子書籍に限定されているこのようなサービスを，紙媒体でも同様に享受することができるようになる．このような検索を実現するための手法の一つとして Locally Likely Arrangement Hashing(LLAH) [2] を用いた文書画像検索がある．LLAH では，文書画像から抽出した特徴点の配置を特徴量とし，特徴量を登録・検索することで，同一文書を取り出す．また，別の手法として，文書画像に対して文字認識を行うことで文書画像をコード化し，そのコードを用いて検索するということも考えられる．

しかし，LLAH の問題点として，検索質問とする文書画像がデータベースの文書と全く同じでなければならないという点がある．つまり，フォントや改行位置，行間など，レイアウトが全く同じでなければならない．これは，コンテンツが全く同じであってもレイアウトが異なるだけで，検索されないということを意味する．例えば，データベースに Web ページの画像が登録されていたとする．ここで，Web ブラウザを用いると，ウィンドウの大きさにより，文章の折り返しが自動で行われる．よって，コンテンツが同じ Web ページの画像であっても，レイアウトが異なるため，LLAH では検索されないという問題点がある．このように，Web 上ではレイアウトが不一致でコンテンツの一致する文書画像が多く存在する．

コンテンツ一致を目的とした検索で最も自然な方法は，文書画像に文字認識を施してコンテンツをコード化し，それを対象として検索するものである．しかし，カメラで撮影した文書画像を入力とする場合，検索に十分な精度で文字を認識することが容易ではないほか，長い処理時間が必要となるという問題もある．文書画像検索の目的が Layered Reading のような実時間性を要求するサービスの提供である場合，これらの問題点はより深刻になる．

本研究の目的は，上記の文書画像検索の諸問題を解決し，コンテンツ一致による文書画像検索法を構築することである．その第一段階として，本研究では対象文書を英文としてシステムを構築する．レイアウトの変動に不変な特徴を取り出すためには，行やカラムなどレイアウトの構成要素に依存しない特徴抽出が必要である．換言すれば，文字や単語といったコンテンツの構成要素のみに依存する特徴抽出が必要となる．この要求を，文字認識のような「重い」処理を施さずに満たすため，本研究では単語の簡易コード化を考える．具体的には，形状特徴を用いて単語画像をいくつかのクラスタに分類し，クラスタ ID の列として文字列をみることによって，文書画像を索引付けする．具体的にはクラスタ ID の n-gram によって索引付けを行う．クラスタ数としては，単語の種類と比べて極めて少数の数百という数を考える．これにより，単語画像が誤分類されることを避け，

処理のロバスト性を得る．

本稿では，まず，関連手法の概要と問題点について説明し，次に提案手法について述べる．その後，今回行った文書画像検索の実用性を検証するための実験とその結果・考察を記し，最後にまとめと今後の課題とする．

2. 従来手法とその問題点

文書画像検索の手法は大きくレイアウト一致に基づくものとコンテンツ一致に基づくものに分類できる．以下，各々について従来法とその問題点を記す．

2.1 レイアウト一致の文書画像検索手法

レイアウト一致の手法として以下の 3 つを紹介する．

Erol らの手法 [3] では，まず単語を囲む矩形を抽出し，矩形ごとに特徴ベクトルを算出する．各単語の特徴ベクトルをデータベースの単語と照らし合わせて単語の投票処理を行う．この時，単語の位置情報も保持しておき，単語の投票結果と周辺単語の位置情報を確認し，文書画像を検索する．

Liu らの手法 [4] では，単語毎に文字をコード化し，近傍の単語コードと組み合わせて特徴量とする．この手法はスキャン画像を対象としたものではなく，カメラベースを想定している．10 万ページのデータベース画像に対して，認識率 95%の精度を得ている．ただし，検索処理には検索質問あたり 4 秒かかるため，実時間性には乏しい．

LLAH [2] もカメラベースの文書画像検索法である．LLAH では，文字や単語の位置を特徴点とし，各特徴点に対して，近傍点を頂点とする図形の面積比を用いて特徴量を計算する．この手法の特徴は大規模データベースから実時間で検索できる点にある．具体的には，2,000 万ページのデータベースに対して，99.2%の精度，50 ms/query の処理時間を達成している．

以上の 3 手法に共通する点は，注目する文字や単語の近傍を，行を跨いで見て特徴抽出することである．このため，レイアウトが崩れると近傍となる単語や文字が変化するため，正しく検索されないという問題点がある．つまり，レイアウト一致の文書画像検索手法では，質問文書と登録文書が改行位置やフォント，行間に至るまで，全く同じレイアウトでなければならない．

世の中にはレイアウトは異なるがコンテンツは同じであるという文書が多数ある．例えば，著作権の切れた文学作品などは，複数の出版社から様々なレイアウトで出版されている．また，Web ブラウザやテキストエディタなどはウィンドウサイズによって自動で折り返しが行われるため，コンテンツは同じでもレイアウトが違う．これらの文書を対象として検索を行うためには，コンテンツの同一性を基準として検索する必要がある．そのためには，上下の行などレイアウトの要素に依存しない特徴抽出をしなければならない

2.2 コンテンツ一致の文書画像検索手法

コンテンツの一致を判定する手法として，まず文字認識を行うことが考えられる．近年ではオープンソースとして OCRplus [5] や tesseract-ocr [6] などがあり，誰でも手軽に利用できるようになっている．これら手法は，スキャナなどで取り込んだ文書を対象として OCR 処理を施し，電子テキストを出力するものである．ただし，一般に文字認識の対象となるカテゴリが多い場合や複雑な特徴を抽出する場合などがあるため，文字認識の計算負荷は少なくない．また，カメラベースの文字認識は発展途上にあるため，十分な認識率が得られているとはいえない．

文書画像の検索だけを目的とするのであれば，より簡便な方法を用いることも可能である．具体的には，すべての文字を正確に認識する必要はなく，簡素化したカテゴリによって認識結果を表現し，検索に役立てることが考えられる．Character Shape Code(CSC) [7] はそのような手法の代表である．CSC は英字を対象とした手法であり，ベースライン，x-height ラインに対して文字がどのような位置を占めるのかを見て，簡易コードを作成する．例えば，a, c など 2 つのラインに挟まれたものは x とし, b, d など，x-height を超えるものは A で表し, g などベースラインの下にも出るものは g とするというように，アルファベットや記号を 15 種類の認識カテゴリに絞り込む．これにより，(1) 文字認識に比べて単純な処理であるため，計算コストを低く抑えられる，(2) 認識のカテゴリ数が少ないため誤認識を生じにくい，という利点を得ることができる．

以上の 2 手法に共通する点は，あくまでもスキャン画像を想定しており，カメラベースは考慮していないことである．具体的には，カメラで取得した文書画像にしばしば生じる文字画像の回転や幾何歪み，接触などには十分対処できない．文書画像検索を用いて Layered Reading のようなサービスを実現する上ではこの問題は深刻となる．加えて，近年のハードウェア，ソフトウェアの進歩はあるものの，OCR は依然として計算コストのかかる処理である．したがって，実時間性を要求するアプリケーションに利用するには困難が伴うという問題もある．

3. 提 案 手 法

3.1 方　　針

従来法における上記の問題点を解決するため，本研究では (1) レイアウトに依存した特徴量を使わずに検索すること，(2) 文字認識と比べて計算コストのかからない処理とすること，(3) カメラベースの入力画像にも十分対応できるロバスト性を持つことの 3 点を目標として，新しい手法を提案する．なお，将来的には多言語での動作を考えているが，現段階では第一ステップとして英文を対象とする．

まず，(1) については，次のように考える．レイアウトが変化しても改行位置を除けば単語の並びは影響を受けない．本研究ではこの点に着目し，単語の並びを特徴とする索引付けを考える．このとき，(2) や (3) を実現するため，CSC と同様の考え方を用いる．すなわち，単語の種類に対して極めて少ないコードを単語画像に割り当てる．これにより，検索には十分ではあるものの安定して抽出できるコードとする．単語を単体で用いるだけでは検索のための特定性が不足するので，単語の n-gram を考え，文書画像の索引付けに用いる．n-gram を構成する際には，連結成分の K 近傍を考慮し，可能な組み合わせを求める．これにより，撮影角度の変動にも対処を試みる．

3.2 処理の流れ

処理の流れを図 1 に示す．

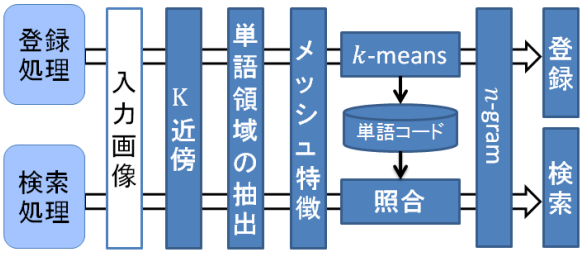

図 1　処理の流れ

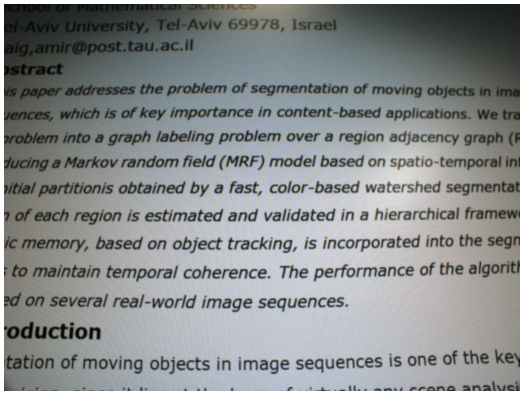

図 2　入 力 画 像

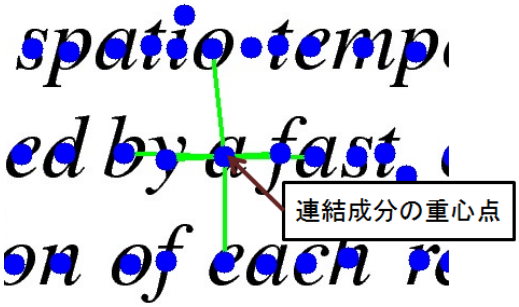

図 3　連結成分重心の K 近傍

まず，図 2 のような文書画像を適応 2 値化した上で黒画素の連結成分を抽出し，その重心を求める．次に，この連結成分の重心を用いて K 近傍を計算することによって，ノードを連結成分，アークを K 近傍関係とするグラフを作成する．K を十分大きく取れば，文字列はこのグラフの部分グラフであると仮定できる．図 3 に例を示す．単語間の空白を考慮しても，通常，$K = 6$ 程度の近傍を考慮すれば，文字列を含めることが可能である．なお，図 4 に示すように，文字に大幅な接触がある場合には，$K = 6$ では不十分となる．ただし，このような状況は支配的ではないため，大きな問題は生じない．

次に，画像処理によって単語領域を求める．方法は極めて単純である．前ステップで得た 2 値画像に対してガウス関数をたたみ込んでぼけた画像を作成し，それを再度適応 2 値化することによって，黒画素の塊を得る．これを単語領域とする．この方法は，LLAH でも用いているものであり，比較的安定して単語領域を得ることができる．

第 3 のステップは，単語画像コード化のための特徴抽出である．本手法ではメッシュ特徴を用いる．図 5 に処理の概要を

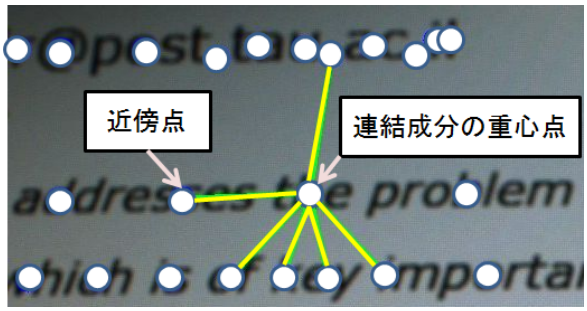

図 4　K 近傍の失敗例

示す．

一般にカメラで撮影した画像には潰れが生じることが多い．このような場合にも安定して特徴を抽出するため，本手法ではまず単語画像そのものではなく，前述の単語領域を求めた際の黒画素の塊を特徴抽出の対象とする．この黒画素の塊を囲む矩形領域に対して，図 5 に示すように $3 \times m$ のマス目を設定する．そして各マスにおいて黒画素の割合を求め，それを b ビット量子化することによってメッシュ特徴とする．現在は $b = 4$ ビット量子化を用いている．単語画像のコード化は以下のように行う．データベース中のすべての単語画像からメッシュ特徴を取り出し，それをクラスタリングする．クラスタ数 s は実験的に定める．

検索処理の際には，上記で得られたクラスタ重心と，検索質問の単語画像から得たメッシュ特徴を照合し，検索質問の単語画像がどのコードに相当するのかを求める．

最後のステップは，n-gram の作成である．このためには単語の並びを求める必要がある．まず，先に求めた連結成分のグラフと単語領域に基づいて，図 6 に示す単語領域のグラフ（単語グラフ）を求める．このグラフは単語領域の重心をノード，単語領域間の近傍関係をアークとするものである．単語領域の近傍関係としては，単語中の連結成分の近傍関係を用いる．すなわち，ある単語領域に含まれる連結成分と，異なる単語領域に含まれる連結成分がアークで結ばれているとき，これら 2 つの単語領域には近傍関係があるとする．

以上の単語グラフを用いると，単語 n-gram は，単語グラフの部分グラフとして求めることができる．このとき，単語列が途中で折れ曲がることはないと仮定し，角度の制約を満たす並びを求める．例を図 7 に示す．この例に示すように，n-gram の最初の単語（図の場合は A）と他の単語の 2 つの単語で構成される角度を見て，すべてが一定の閾値以下（この場合は 0.15 radian 未満）であれば n-gram として認定する．

処理例を図 8 に示す．この図は単語 n-gram として採用されたアークを図示したものである．この例では，上記の処理により概ね単語の並びが正しく取り出されている．ただし，角度の制約によって除外された重心（この場合はカンマに相当）があるほか，“partitionis region based temporal” のように本来の単語列とは異なる方向の組み合わせも得られることがある．単語の方向を考えるとこのような例を排除することはそれほど難しくないが，数が多くないために，本手法では，そのまますべ

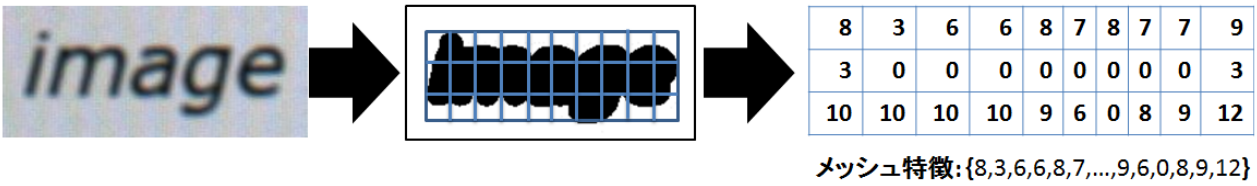

図 5　メッシュ特徴

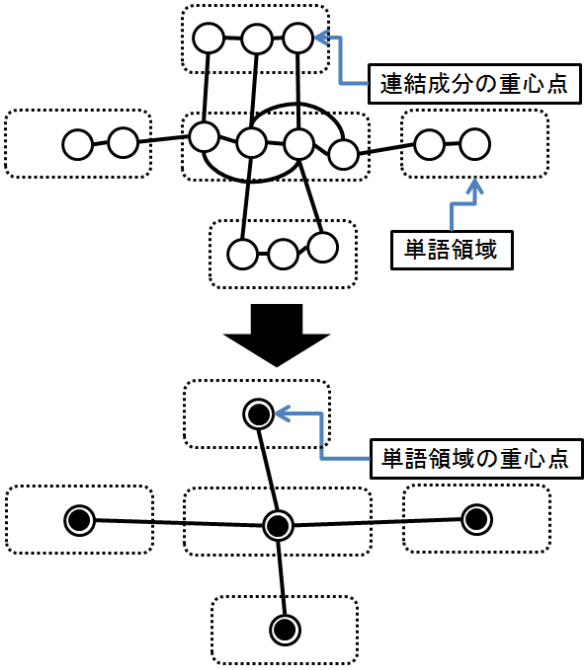

図 6　単語グラフ

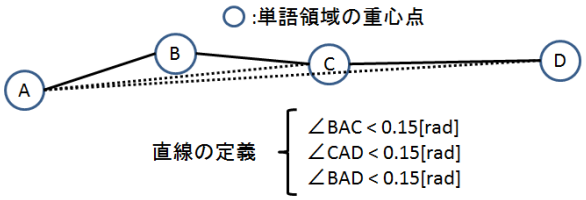

図 7　n-gram

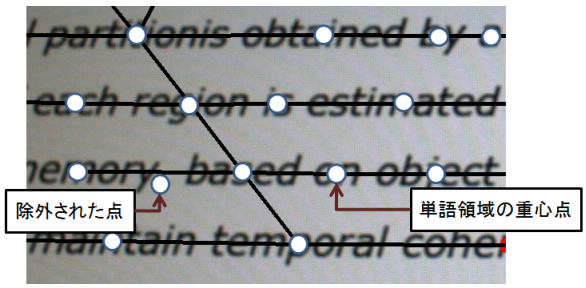

図 8　単語列の抽出

てを n-gram として採用する．

図 1 に示すように，登録処理では，単語 n-gram の抽出が終了するとそれをデータベースに登録する．n-gram は単語コードの n 個の並び $(x_1, x_2, ..., x_n)$ として表現できるので，この並びをキーとして文書 ID を値とするハッシュ表に登録する．ハッシュ関数としては，

$$H = \sum_{i=1}^{n} x_i s^{i-1} \tag{1}$$

を用いる．このときハッシュテーブルのサイズは s^n である．ハッシュ表に登録する際に衝突が生じると，データをチェイン法で記録しておく．ただし，多くの衝突が生じることはその n-gram が十分な特定性を持たないことを意味するため，チェイン法で登録されるリストの長さが c 以上となると，そのリストを削除した上で，以後の登録を受け付けないものとする．

検索処理は次のようになる．メッシュ特徴を抽出する際に単語の輪郭だけを用いているため，単語の形状によっては，検索質問画像から得た単語画像が正しいクラスタに対応付かない場合が考えられる．この問題に対処するため，本手法では，検索質問の単語画像のクラスタを 1 つに決めてしまうのではなく，メッシュ特徴が近いものから r 個の候補を用いる．すなわち，検索質問からは n 単語の各々の並びに対して，r^n 個の n-gram を生成して検索処理に用いる．

具体的な利用法は以下の通りである．各 n-gram についてハッシュ表を参照し，登録されている文書 ID を検索する．そして各文書 ID に対して投票処理を行う．最終的に最大得票となった画像を検索結果として出力する．

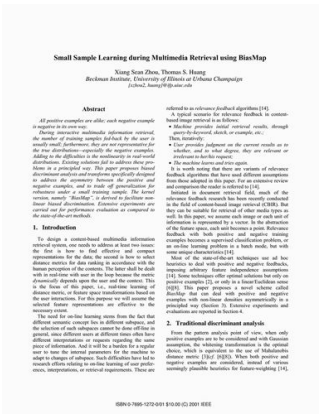

図 9　データベース画像例

図 10　クエリ画像例

4. 実　　験

提案手法の有効性を検証するために 2 種類の実験を行った．まず，提案手法を用いて撮影文書画像から n-gram を抽出し，検索精度を検証した．次に，対比実験として OCR を用い，検索精度と処理時間を比較した．

(a) Times　(b) Century　(c) Arial　(d) Verdana

図 11　各種フォント

4.1 実　験　1

実験 1 は次の条件下で行った．まず，データベース中の文書 10,000 ページからレイアウトを変更した 25 のクエリ文書を作成した．データベース文書の例を図 9 に，クエリ文書の例を図 10 に示す．データベースの文書画像は 1 ページごとに文書 ID を付与している．このとき，クエリには図を含まないことを仮定している．また，クエリがデータベースの複数のページにまたがらないことも仮定している．

クエリを作成する際には，Times New Roman(以下, Times), Century，Arial，Verdana の 4 種類のフォントを用いた．例をそれぞれ図 11(a)，図 11(b)，図 11(c)，図 11(d) に示す．フォントは大きく分けて “セリフ” と “サンセリフ” の 2 つに分類される．セリフとは，アルファベットをデザインするときに線の端に図 11 中の丸で囲まれた部分のような “飾り” が付いているフォントのことである．サンセリフは逆に飾りのない文字である．Times や Century はセリフ，Arial や Verdana はサンセリフである．データベース中の文書には Times が用いられている．

次に，作成した各クエリ文書をディスプレイに表示しほぼ正面から 3 回撮影した．これによって得られる $25 \times 4 \times 3 = 300$ 枚の画像をクエリ画像とした．

パラメータの値は以下の通りである．K 近傍を $K = 6$，メッシュを 3×10，n-gram を $n = 4$ に固定した．また，単語のクラスタ数 s を $s \in \{128, 256, 512\}$，検索質問の単語画像に与える単語コードの数 r を $r \in \{1, 2, 3, 4, 5, 6, 7\}$，$n$-gram のハッシュ衝突回数上限値 c を $c \in \{1, 2, 4, 8, 16\}$ としてすべての組み合わせで性能を評価した．

4.2 実　験　2

実験 2 では OCR を用いて対比実験を行った．OCR としてはオープンソースである tesseract-ocr を用いた．OCR を用いた手法では，OCR により文書をコード化し，単語列を得た．そして，単語列の n-gram をすべて求め，ハッシュに格納した．提案手法と条件を合わせるため，$n = 4$ で実験を行った．このとき，提案手法とハッシュテーブルのサイズが等しくなるように調整した．こちらも提案手法同様に n-gram のハッシュ衝突回数の上限値 c を変動させた．処理時間の比較では，データベースの読み込み時間を処理時間に含めずに照合の時間だけを計測した．

表 1　フォントごとの検索精度 [%]

	フォント				
	セリフ体		サンセリフ体		
	Times	Century	Arial	Verdana	合計
提案手法	100	98.7	85.3	90.7	93.7
OCR	97.3	98.7	100	100	99.00

5. 結果と考察

5.1 実　験　1

データベースの n-gram は約 500 万個であった．$s = 256$ の場合を例にすると，クラスタ番号を組み合わせた n-gram の種類は $(2^8)^4 = 2^{32}$ であり，平均衝突回数が 1.32 であった．衝突回数が大きい原因は，今回の実験で用いたデータベース画像が全て同じ学会誌の論文であったために，全ての画像から同一の出典表記があったことが挙げられる．また，1 枚の文書画像から抽出される n-gram が非常に多い場合があった．これは図を構成する細かい点が一つの単語領域として抽出されている場合などであった．1 枚のクエリ画像から生成される n-gram の数はおよそ数十〜数百程度であった．

最も良い正解率を示した条件設定はクラスタ数 $s = 256$，コード数 $r = 5$，上限値 $c = 1$ のときであり，検索精度は 93.7%となった．表 1 に，クエリのフォント毎の検索精度を示す．フォントによって検索精度に偏りがあることが分かる．検索精度が高いのは，セリフ体の文書画像であった．これは，データベースの文書が Times フォントで書かれているためである．

図 12 に登録画像中の単語 “image” を，図 13 に撮影画像 (フォント Courier) 中の単語 “image” を図 14 に撮影画像 (フォント Verdana) 中の単語 “image” をそれぞれ示す．図 13 と図 14 の “image” は一見似た形に見えるが，図 14 の “image” は縦の長さが短いため，横方向の分割に差が出る．また，文字の “飾り” の有無によって矩形の大きさや形が変わる．図 13 を見ると，飾り部分がぼけることで隙間が埋められている点も相違している．フォントの相違が検索結果を低下させる原因となった．この問題に対しては，データベース中に様々なフォントの画像を登録しておくことが考えられる．データは大量となるものの，ハッシュ表を用いれば検索時間にはさほど影響は出ないものと考えられる．

$r = 5$ のときに最も高い検索精度を得たということは，クラスタリング結果を 5 位まで見て正しいクラスタに分類される単語が多いということである．換言すれば，単語のクラスタリング精度は高くないということである．クラスタリングの精度を上げる簡単な方法は，クラスタ数 s を少なくすることであるが，これを行うと特定性が低下するためハッシュ表での衝突が増え，検索精度が低下する．クラスタリングの精度を改善するためには，輪郭形状を見る現在の方式を変更し，単語の内部の構造も参照する必要があろう．ただし，この際には，ロバスト性を失わない仕組みと対で用いる必要がある．

衝突回数の上限 c の値が小さいときに最適になったということは，検索に有用ではない n-gram が多数生成されていること

図 12　登録画像中の “image”

図 13　撮影画像 (フォント C) 中の “image”

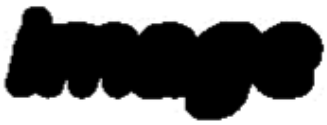

図 14　撮影画像 (フォント V) 中の “image”

を示唆している．単語クラスタリングの精度を向上させることによって，この問題点も改善されると考えられる．

5.2 実　験　2

OCR を用いた場合，データベース文書画像から約 760 万個の n-gram が生成された．OCR では，n-gram とハッシュテーブルのサイズを提案手法と同等とした．検索精度が最も高かったのは，$c = 8$ のときであり，99.0% を得た．同様に，フォント毎の検索精度を表 1 に示す．OCR を用いた場合はフォントによる偏りは殆どなかった．

OCR を用いて検索に失敗する原因は 2 つである．1 つは，文字認識に失敗している場合，もう 1 つは，単語の区切りの認定に失敗し，結果として n-gram が正しく生成できなかった場合である．

提案手法に比べて OCR を用いた手法の検索精度が高い理由は以下である．検索成功時の平均投票数は提案手法が 13.3 票なのに対し，OCR を用いた手法では 27.1 票と，およそ倍になった．この原因は，データベース文書において改行位置を識別できているかどうかの違いである．提案手法では改行を跨いで単語 n-gram を作成することはできない．一方，OCR では改行が認識され，文字列が連結されるため，改行があっても n-gram を生成することができる．ただし，これは段落がすべてカメラで撮影された場合に成り立つ利点である．段落の一部分だけが撮影された場合には，この性質が悪影響を及ぼすと考えられる．

逆に OCR を用いて認識できていないが，提案手法では認識できているというクエリ画像もある．その多くは図を文字として誤認識したものである．図の中には多くの連結成分を含むものがあり，それを文字として誤認識すると大量の無意味な n-gram が生成される．一方，クエリ画像の場合はディスプレイのノイズや照明条件などにより，様々なノイズの連結成分が現れる．このノイズによる n-gram が，図から OCR により生成された n-gram と結びつき，誤検索が生じた．

検索質問あたりの処理時間を比較すると，提案手法が最も認識率が高いパラメータで 413[ms] であったのに対し，OCR では 1,712[ms] であった．提案手法では処理時間全体の約 1/4 が単語コードの組み合わせに費やされている．最高認識率を得た $r = 5$ では，1 つの n-gram の領域に対して，$5^4 = 625$ 個の単語コードの組み合わせを処理に用いている．r を小さくすれば処理時間は早くなるものの，検索精度の低下も招く．例を挙げると，$r = 4$ のとき検索精度 92.67%で処理時間 348[ms]，$r = 3$ のときは検索精度 90.67%で処理時間 315[ms] となった．一方，OCR では処理時間の大部分を OCR 処理が占めている．

最後にハッシュテーブルの使用率を比較する．提案手法も OCR を用いた比較手法も，ハッシュテーブルのサイズを 256^4 とした．この時，提案手法では実際に使用しているビンの数は 0.08%であり，比較手法では 0.15%であった．一方，n-gram の登録数は提案手法が 500 万個，OCR が 760 万個であり，提案手法の方が少ない．このことから，提案手法の n-gram が OCR で得られるものに比べてあまり分散していないことがわかる．これはハッシュの衝突回数の違いにも現れている．提案手法が 1.32 であることに対し，比較手法は 1.15 と低い．以上より，提案手法の特徴抽出には改善の余地があるといえる．

6. ま　と　め

表示デバイスによらず文書画像検索を行うにはコンテンツ一致の基準による検索手法が必要となる．また，様々なサービスに利用可能とするためには，実時間性も求められる．このような 2 つの要求を満たす手法として，本稿では，単語クラスタの n-gram に基づく検索法を提案した．この手法の特徴は，単語クラスタ数を抑えることにより安定性を得つつ，n-gram として組み合わせることによって検索に必要な特定性を得る点にある．また，画像撮影時の変動に対処するため，単語の内部形状を塗りつぶした画像からメッシュ特徴を抽出し，処理に用いた．これらの単純な処理を組み合わせることによって，OCR を用いた場合の 1/4 の処理時間で検索を行うことが可能となった．このときの検索精度は 93.7%であった．

今後の課題には，特徴抽出や索引付けの改良によって，さらに検索精度を高めることや，より大量の実験サンプルを用いて実用性を評価することなどがある．

謝　　辞

本研究の一部は日本学術振興会科学研究費補助金基盤研究 (B)(22300062) の補助による．

文　　献

[1] http://84dialog.blogspot.com/2010/03/layered-reading.html, 2011
[2] 中居, 黄瀬, 岩村：“デジタルカメラによる文書画像検索―1 万ページから 0.1 秒で検索する―”, 情報科学技術レターズ, **4**, pp. 133–136 (2005).
[3] B. Erol, E. Antúnez, and J.J. Hull, “HOTPAPER: multimedia interaction with paper using mobile phones,” Proceeding of the 16th ACM international conference on Multimedia, pp.399-408, 2008.
[4] X. Liu and D. Doermann, “Mobile Retriever: access to digital documents from their physical source,” Int. J. Doc. Anal. Recognit.,vol.11,pp.19-27,Sept. 2008.
[5] http://code.google.com/p/ocropus/
[6] http://code.google.com/p/tesseract-ocr/
[7] H. Bunke and P. S. P. Wang Eds.: “Handbook of Character Recognition and Document Image Analysis”, Vol. 9, World Scientific Pub Co Inc (1997).