レイアウト非依存な実時間カメラベース文字認識

岩村　雅一† 　辻　 智彦† 　堀松　 晃† 　黄瀬　浩一†

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

あらまし　カメラを用いる文字認識の実用化を見据えると，(1) 実時間処理が可能である，(2) 射影歪みに頑健である，(3) 文字の配置に依らない認識が可能であるという 3 つの要件を満たす実用的なカメラベース文字認識技術が求められる．本論文では，前述の 3 要件を全て満たすカメラベース文字認識手法を提案する．提案手法は，単純だが効果的な処理を組み合わせることにより，安価な web カメラと接続したノートパソコン上で実時間で動作する．実時間処理の実現のために，提案手法では幾何学的不変量の計算原理を通常とは異なる方法で用いた．その結果，認識処理に必要な計算量が大幅な削減され，サーバーを利用した場合には 1 秒間に 200〜250 文字程度の認識が可能になった．

キーワード　カメラベース文字認識，不変座標系，ハッシュ法，姿勢推定

Real-Time Camera-Based Character Recognition Free from Layout Constraints

Masakazu IWAMURA†, Tomohiko TSUJI†, Akira HORIMTSU†, and Koichi KISE†

† Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuencho, Naka, Sakai, 599-8531 Japan
E-mail: †{masa,kise}@cs.osakafu-u.ac.jp, †tsuji@m.cs.osakafu-u.ac.jp

Abstract　In order to realize an application of camera-based character recognition, a character recognition technique which is (1)ready for real-time processing, (2)robust to perspective distortion, and (3)free from layout constraints, is required. In this paper, a camera-based character recognition method which satisfy the three requirements is proposed. As a result of a combination of simple but efficient techniques, the proposed method enables us to execute camera-based character recognition in real-time even on a laptop PC with a cheap web camera. Real-time processing is realized using a principle of geometric invariants in a different manner than usual. This drastically reduced computational cost and achieved fast recognition of around 200 to 250 characters per second on a server.

Key words　Camera-Based Characters Recognition, Invariant Coordinate Systems, Hashing, Pose Estimation

1. はじめに

カメラを用いる文字認識は様々な応用を見込めることから，近年注目を集めている．有望な応用としては，カメラと文字認識装置，翻訳装置を組み合わせた「翻訳カメラ」がある [1]．また，カメラで写した文字を認識し，音声に変換することで視覚障害者に伝える応用も考えられる．さらに別の応用としては，カメラに写る全ての文字を認識し，その中から事前に登録しておいた有用な情報のみを利用者に提供することも考えられる．視覚障害者の中には文字を見つけるのが困難な人もいるため,「機械による視覚」ともいえるこの応用は非常に有用である．

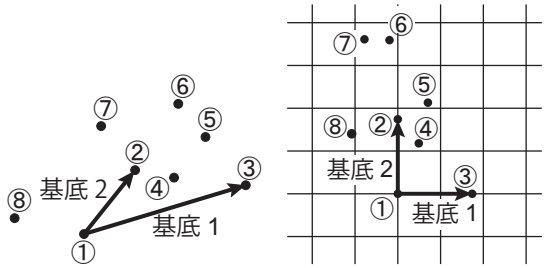
上記のような応用を実現するためには，(1) 実時間処理が可能である，(2) 射影歪みに頑健である，(3) 文字の配置に依らない認識が可能であるという 3 つの要件を満たす実用的なカメラベース文字認識技術が必要である．

前述の (1) と (2) の要件を満足する手法として，Myers らの手法 [2] がある．この手法は文字行単位で射影歪みを補正するので，文字行を成さない文字は認識できない．また，回転した文字にも対応していない．そのため，図 1 のような対象は認識できず，(3) の要件を満たさない．

一方，前述の (2) と (3) の要件を満たす手法として，Kusachi らや Li らは文字を 1 文字ずつ認識する手法を提案している [3], [4]．これらの手法は処理に時間がかかり，(1) の実時間性の要件を満たさない．このように，3 要件を同時に満たす手法はこれまで提案されてこなかった．

本論文では，前述の 3 要件を満たす単純だが効果的な文字認識手法を提案する．提案手法の文字切り出しには適応二値化を用い，輪郭抽出を行う．提案手法の文字認識には，Geometric Hashing(以下，GH) を改良して用い

図 1　文字行が直線でないようにデザインされた文書の例.



(a) 基底の決定.　(b) 不変座標系への射影.

図 2　Geometric Hashing の不変座標系.

る．アフィン変換に対応した GH の計算量は P を点数としたとき，$O(P^4)$ であるが，提案手法では不変量の計算原理を利用して $O(P^2)$ にまで削減可能である．また，認識と同時に多数決原理を利用した姿勢推定を行うことで，誤認識の少ない頑健で実用的な認識が可能になる．

添付資料中の動画に示すように，提案手法は安価な web カメラと接続したノートパソコン上で動作する．

1.1　問題設定と解決方法

本論文の問題設定について確認しておく．過去の同様の研究に習い，簡単のために，白色の背景に黒色の文字が書かれていることを想定する．文字画像はカメラで撮影するので，射影歪み，ぼけや解像度低下の影響を受けるが，文字の連結成分 (注1) は簡単な処理で切り出せるとする．また，全ての文字は同一平面上に存在するとする．

本論文で解決する問題は，(i) 切り出された連結成分の高速な認識，(ii) 認識の頑健性向上，(iii)「i」や「j」のように 2 つ以上の連結成分から成る文字 (分離文字) の認識，の 3 つである．

(i) については，GH を連結成分の照合に適応し，幾何学的不変量の計算原理を利用して高速化する方法を 2. 節で示す．(ii) については，連結成分の姿勢を考慮した認識方法を 4. 2. 4 節で述べる．(iii) については，3. 節で述べる一般化ハフ変換に類似のアイディアで解決する．

2.　連結成分の高速な認識

2.1　Geometric Hashing

GH は幾何歪みを受けた画像を不変座標系を用いて記述し，認識する手法である．提案手法の説明のために，まずアフィン変換に対応した GH を簡単に説明する．

2.1.1　登 録 処 理

登録する参照画像から特徴点が抽出されているとする．まず，図 2(a) に示すように，全特徴点から 3 点をランダムに選び，2 本のベクトルを作成する．次に，図 2(b) のように，この 2 本のベクトルを基底とする新しい座標系を作成する．この座標系は，画像がアフィン変換を受けても同様に作成できるため，アフィン不変座標系である．このアフィン不変座標系を図 2(b) のように格子状に区切ると，各領域は 2 次元ハッシュテーブルのビンになる．各特徴点について，特徴点が存在するビンに画像の番号と基底の組の通し番号を登録する．この処理を全ての作成可能な基底に対して実行すると，1 枚の参照画像の登録が終了する．全ての参照画像を登録して登録処理が終了する．

参照画像 1 枚の登録に要する計算量は $O(P^4)$ である．内訳は，アフィン不変座標系の作成に $O(P^3)$，ハッシュの参照や投票処理等に $O(P)$ である．

2.1.2　認 識 処 理

認識対象の画像 (質問画像) から特徴点が抽出されているとする．まず，登録処理と同様に特徴点を 3 点選び，図 2(b) のような不変座標系を生成する．このアフィン不変座標系は登録時に格子状に区切られ，ハッシュテーブルになっているので，特徴点が存在するビンから，登録されている画像の番号と基底の組の通し番号を取り出して，2 次元の投票テーブルに投票する．この処理を全ての作成可能な基底に対して実行し，最大の投票数を得た画像の番号と基底の組の通し番号を決定する．そして，画像の番号を認識結果として出力する．ただし，全ての基底を処理する前に結果が明らかになったときには処理を途中で終了することができる．

質問画像 1 枚の認識に要する計算量は $O(P^4)$ である．内訳は登録処理と同じである．

2.2　Geometric Hashing の改良

2.2.1　問題設定の違い

GH を基にして，連結成分を高速に認識する手法を提案する．提案手法について述べる前に，GH と提案手法の問題設定の違いについて述べておく．

GH が解いている問題は，特徴点の配置のみから対象を同定する問題である．つまり，特徴点がどのような対象からどのように抽出されたかを考慮しない．それに対して本論文で考える問題では，特徴点は連結成分の輪郭の画素とわかっており，連結成分そのものも与えられる．つまり，提案手法では GH より多くの情報が与えられ，それを積極的に利用することで高速化が可能になる．

(注1)：画像中で黒画素が隣接し，一塊になっているものを指す．

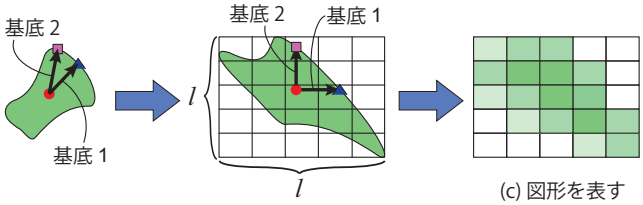


図 3　特徴ベクトルの計算方法．$k = l \times l$ としたとき，k 個の部分領域のヒストグラムを利用して k 次元の特徴ベクトルを作成する．便宜上，黒画素を緑色で表している．

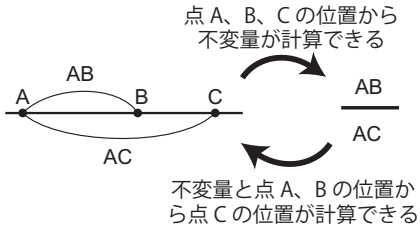


図 4　点の位置とアフィン不変量の関係．

以降の処理は連結成分の認識に特化するため，「画像番号」のことを「連結成分番号」と呼ぶことにする．また，提案手法では原則として図形の一番外側の輪郭の画素を全て特徴点とする．

2.2.2　図形の特徴量の使用

GH では特徴点の位置そのものを特徴量とみなせたが，提案手法では連結成分の形状を表す特徴ベクトルを新たに作成して用いる．

具体的な特徴ベクトルの作成方法は以下の通りである．まず 2.1 節で述べた手順で図 2(b) のような不変座標系を作成する．この結果，図 3(a) の図形が図 3(b) のように正規化されたとする．このとき，$l \times l$ 個の均一な部分領域 ($k \equiv l \times l$ とする) を設定し，図 3(c) のような黒画素数のヒストグラムを計算することができる．最後にヒストグラムの合計が 1 になるように正規化し，k 次元特徴ベクトルとする．この特徴ベクトルは，同一の基底の組が選択される限り，理論上，アフィン変換の影響を受けない．本論文では，予備実験により，$l = 4$ とした．

画像から計算した特徴ベクトルを用いる理由は，画像の特徴のほうが表現力が高いと考えられるからである．本論文では黒画素数の正規化ヒストグラムを特徴ベクトルとしたが，将来的にはスキャナで撮像した文字認識で使用される既存の正規化手法や特徴量を使用することも可能である．

2.2.3　計算量の削減

GH の欠点は膨大な計算量である．アフィン変換に対応した場合について考えると，認識処理に必要な計算量は点数 P に対して $O(P^4)$ である．すなわち，$P = 100$ 点とすれば $O(100,000,000)$ もの計算が必要になるため，実時間アプリケーションに使用することは現実的でない．

計算量を削減するため，GH の計算量が膨大である理由を考える [5]．GH では認識が成功するために，登録処理で用いた基底の組が認識処理でも計算される必要がある．しかし，基底の組が一致するかどうかは計算して照合してみるまでわからないため，結局全て (もしくは多数) の基底を計算することになってしまう．つまり，登録処理と認識処理で同一の基底を効率良く計算することができれば計算量を削減することができる．提案手法では，幾何学的不変量の計算原理を通常とは異なる方法で用いることにより，これを実現する．

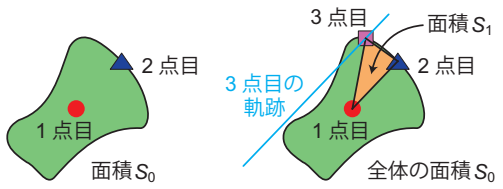
提案手法の具体的な手順を示す前に，図 4 の例を用いて特徴点を一意に選択する原理を説明する．図 4 のように 1 直線上に 3 点 A，B，C が与えられたとき，AB/AC はアフィン変換を受けても変化しない不変量である．このように点の座標から不変量の値を計算することが通常行われる．これに対して提案手法では，不変量の値と 2 点 A，B が与えられたときに点 C の座標を求める．点 C が点 A，B と同一直線上にあるとすると，点 C の位置は点 A の左側にある場合と図 4 のように点 B の右側にある場合の両方が考えられるが，「直線上に A，B，C の順番に並ぶように点 C を定める」といった具合に点 C の決定方法を事前に定めておけば一意に定めることが可能になる．この原理を一般化して記述すると，「n 点の座標から計算した不変量の値と $n-1$ 点の座標が与えられれば，残る 1 点 (n 点目) を計算することができる」となる．このように基底の作成に用いる特徴点を一意に決定できれば計算量を削減できる．

提案手法の特徴点の決定方法について具体的に述べる．まず，アフィン変換では図形の重心が保存されるため，重心を 1 点目の特徴点とする(注2)．2 点目は GH と同様に，ランダムに選択する．3 点目は，ここまでに得られた 2 点と前述した不変量の性質を用いて一意に定める．このようにすれば，1 点目と 3 点目は一意に決定できるので，計算量を $O(P^4)$ から $O(P^2)$ に削減することができる．

ただし，上記の特徴点決定方法を用いたとしても 3 点が一直線上に存在してしまうため，不変座標系を作成するのに必要な一次独立な基底を 2 本作成することができない．そこで以下では，図 5(a) のように面積が S_0 である図形を考え，3 点目を一意に決定できる上記とは別の方法を例示する．

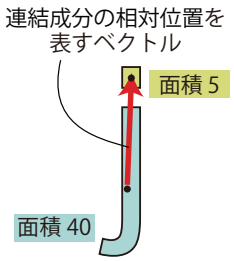
3 点目の特徴点が仮に図 5(b) のように与えられたとする．ここで 3 点が作る三角形の面積を S_1 とすると，S_1/S_0 はアフィン不変量になる．したがって，S_1/S_0 があらかじめ定めた特定の値になるように 3 点目を定めればよい．S_1/S_0 の値としては，特定の値以外に最大値を用いることも可能である．3 点目を定める際に時計回りや反時計回りなどの情報を使って点を一意に定めること

(注2)：重心が輪郭上になる保証はないが，特徴点が輪郭以外に存在しても後の処理に影響はない．



(a) 元の図形.　(b) 3 点目の定め方.

図 5　面積比を用いて 3 点目を一意に決める方法の例.



①	②	③	④	⑤
J	j	↑	40	5
·	j	↓	5	40
I	i	↑	25	5
·	i	↓	5	25

(a) 連結成分の記述方法　(b) 分離文字テーブル

図 6　分離文字の記述方法．分離文字テーブルの要素は左から，①連結成分の形状 (連結成分番号)，②元の文字，③連結成分の相対位置，④自連結成分の面積，⑤組になるべき連結成分の面積である．

ができる．なお，S_1 が一定になるように 3 点目を定めることを考えると，取り得る 3 点目の軌跡は図 5(b) のように 1 点目と 2 点目を通る線分と平行な直線になる．したがって，この直線と図形との交点を 3 点目に定めればよく，簡便に計算可能である．交点が複数ある場合は 2 点目から遠いほうを選ぶといった選択方法も可能である．本論文では，面積が最大になる方法を実験で用いた．

3.　分離文字の認識方法

前節では単一の連結成分で構成される文字を高速に認識する方法を述べた．本節ではその結果を利用することで,「i」や「j」のように 2 つ以上の連結成分から成る分離文字の認識方法を述べる．

まず，分離文字に対処するため，参照画像の登録時に画像内の連結成分数を調べる．そして，2 つ以上の連結成分を含む参照画像に対しては，各連結成分を別の文字のように扱い，別々に登録するとともに，図 6 の分離文字テーブルを作成する．このテーブルは分離文字の各連結成分の位置と大きさの関係を記したもので，認識時に所定の位置に所定の連結成分があるかどうかを調べることで分離文字の認識が可能になる．

ただし，Arial の場合について考えてみると,「i」の下の連結成分は「I」(大文字のアイ) や「l」(小文字のエル) と同じ形状のため，外見では区別がつかない．そのため,「i」を正しく認識するためには,「I」や「l」のように同形である全ての連結成分に対して,「i」であるか調べる必要が生じる．そして,「i」の上の連結成分が所定の位置に存在すれば「i」と認識し，そうでなければ「I」や「l」と認識することになる．

表 1　アフィン歪みを受けた場合の類似文字のリスト．同じ箱に入っている文字はアフィン歪みを受けた場合に認識が難しいので，同一のクラスとした．

0 O o	6 9	C c	I l	S s	u n
W w	X x	N Z z	p d	q b	7 L V v

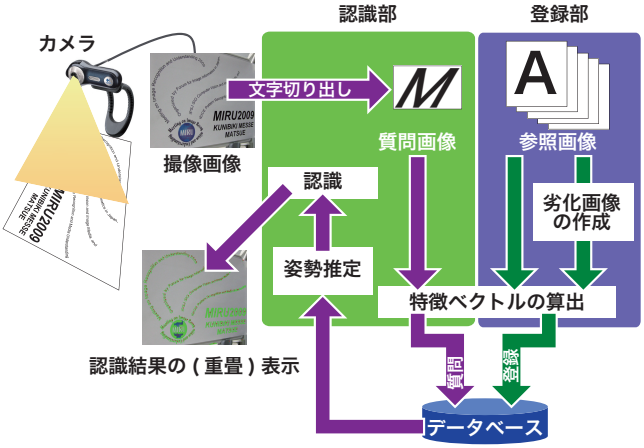


図 7　提案システムの概要.

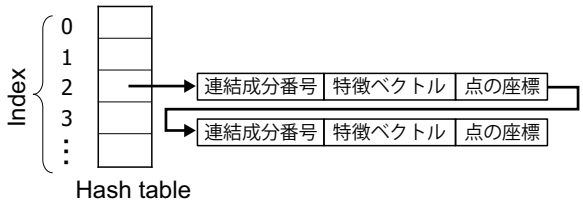


図 8　提案手法において連結成分番号を保持するハッシュ表.

この処理を実現するために，同形である全ての連結成分が同じ連結成分番号を持つように学習した．すなわち，参照画像は 1 枚ずつ登録することとし，登録前に類似の連結成分が登録されていないかを調べた．より具体的に書くと，参照画像の登録前に作成中のデータベースを用いて認識し，認識結果に同形の参照画像が含まれていた場合は同じ連結成分番号を割り振った．理想的には，前述の「i」の下の連結成分や「I」,「l」は全て同じ連結成分番号を持つはずなのだが，この方法は後述する生成型学習との相性が悪く，同形になるべき字種が同形にならない場合があった．そのため，本論文では表 1 に示す類似文字リストに基き，手動で設定した．

4.　提案システム

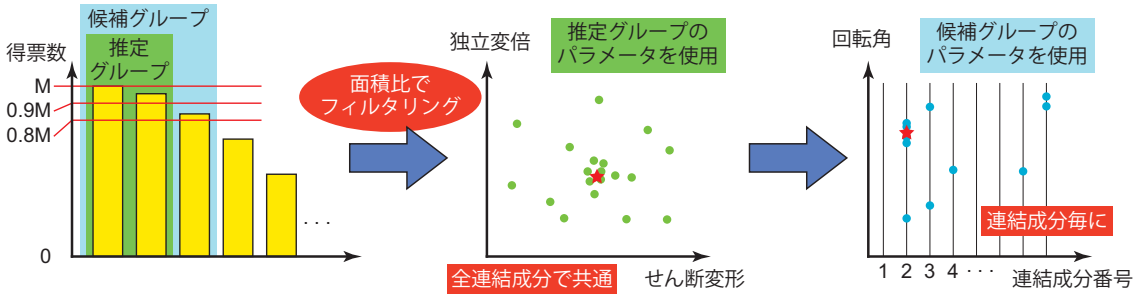
提案システムの概要を図 7 に示す．提案システムは画像登録部と画像認識部から成る．それぞれの詳細について，以下で説明する．

4.1　画像登録部

画像登録部では，参照画像をデータベースに登録する．参照画像は二値画像とする．

4.1.1　劣化画像の生成

撮影時のピンぼけや解像度の低下に対処するため，生



(a) 連結成分毎の得票数（得票数順にソート後）　(b) 紙面の姿勢推定　(c) 連結成分の姿勢推定と種類の決定

図 9　連結成分の認識と姿勢推定.

成型学習法 [6] を用いて劣化した参照画像を生成する．本論文では，ガウスぼかしの重畳を 3 段階 (適用なしを含む) と解像度の低下を 3 段階 (適用なしを含む) の合計 9 段階で行い，元の参照画像を含めて 9 枚の参照画像を得た．生成された劣化画像は再び二値化し，以後は元の参照画像と同様に扱う．

4.1.2　特徴ベクトルの生成

2.2.2 節で述べた方法で特徴ベクトルを作成する．ただし，実際には 2.2.2 節の方法で作成したベクトルを以下の方法で組み合わせて使用する．

特徴点が 3 点あれば，そのうち 1 点を原点とすることにより，2 本の基底ベクトルが計算でき，図 3(b) の不変座標系が計算できる．このとき，特徴点の原点を入れ替えることで，基底の組は 3 通り作成することができる．すなわち，同一の 3 点から 3 本の k 次元特徴ベクトルが計算できる．それらを単純に結合することで得られる $3k$ 次元の特徴ベクトルを提案手法で使用する．結合する順番は特徴点が選択された順序に基づいて一意に決定する．

4.1.3　データベースへの登録

提案手法では，連結成分番号を図 8 に示すハッシュテーブルで保持している．ハッシュテーブルには，連結成分番号，特徴ベクトル，基底の作成に用いた特徴点 3 点の座標を一組にして格納し，衝突が起こる場合はリストで連結する．特徴点の座標を登録しておくのは，認識の際の姿勢推定で利用するためである．提案手法では特徴ベクトルを使用するため，ハッシュテーブルは GH のような 2 次元ではなく，1 次元になる．

ハッシュのインデックス H_{index} は次式で計算する．

$$H_{\mathrm{index}} = \left(\sum_{i=1}^{3k} D^{i-1} r_i \right) \bmod H_{\mathrm{size}} \qquad (1)$$

ここで H_{size} はハッシュテーブルの大きさであり，$2^{19}-1$ とした．また，r_i は特徴ベクトルの i 番目の要素を D 段階に量子化した値で，予備実験により $D=3$ とした．

4.2　画像認識部

4.2.1　画像の取得

画像はデジタルカメラや web カメラで静止画ないし動画として取得する．動画として撮像した場合はフレーム毎に分解して，複数の静止画として扱う．得られた各画像を質問画像と呼び，以下の処理で用いる．

4.2.2　連結成分の切り出し

得られた画像から文字画像を切り出す．まず画像に対して，適応二値化を施す．すなわち，注目画素の輝度が近傍領域の平均輝度より明るいか暗いかに応じて，注目画素を白または黒と決定する．本論文では，注目画素を中心とする一辺が n の正方形を近傍領域とし，$n=101$ とした．

次に，連結成分を順に切り出し，以下で述べる認識処理の対象とする．ただし，連結成分の面積 (ピクセル数) が閾値以下であればノイズとみなして認識対象から除外する．本論文では閾値を 32 とした．

4.2.3　特徴ベクトルの生成

連結成分から特徴ベクトルを生成する．この処理は 4.1.2 で述べた処理と基本的に同じである．唯一異なるのは，作成可能な全ての特徴ベクトルを計算するのではなく，あらかじめ定める S 個に限定することである．S と性能の関係については 5.2 節の実験で議論する．

4.2.4　認識と姿勢推定

前節で述べた特徴ベクトルを用いることで，図 8 のハッシュテーブルから連結成分番号と特徴点 3 点の座標を取得することができる．こうして得られた情報は，質問画像の仮の認識結果であり，多数の誤りを含んでいる．以下では文献 [7] に類似の多数決原理を段階的に用いることで正しい認識結果に集約する．すなわち，図 9 に示すように，最初に紙面の姿勢推定を行い，続いて連結成分毎に認識と姿勢推定を行う．

まず，質問画像の特徴点と参照画像の特徴点の対応関係から，質問画像中の連結成分の姿勢がアフィン変換行列として推定される．この中には別の連結成分に対応付いて求められた誤ったアフィン変換行列が含まれているため，図 9(a) のような連結成分番号に対する重み付き投票を行い，信頼できるものを選定する．重みを用いるのは，特徴点が多い連結成分は得票が不公平に多くなると考えられるためである．i 番目の連結成分に登録されている特徴点数 (外側の輪郭の長さ) を N_i としたとき，$\frac{1}{\sqrt{N_i}}$ の重みを各投票度数に対して掛ける．

重み付き投票によって得られた最大の得票数 (M とお

く) を基準にして，2 つのグループを定義する．1 つ目は，得票数が $0.9M$ 以上である参照画像の連結成分のグループで,「推定グループ」と呼ぶ．2 つ目は，得票数が $0.8M$ 以上のグループで「候補グループ」と呼ぶ．これらは質問画像の連結成分毎に定める．

次に，紙面の姿勢を推定する．本論文では全ての文字は同一平面 (紙面) 上に存在すると仮定している．この場合，アフィン変換行列から計算される 4 つのアフィン変換パラメータのうち，せん断変形と独立変倍のパラメータは全ての連結成分で共通になるはずである．そこで文献 [7] と同様に，図 9(b) のような 2 次元空間での密度推定を利用して，尤もらしいパラメータの組を推定する．ここでは「推定グループ」に属する連結成分のアフィン変換行列を前述の 2 次元空間にプロットする．プロットされた点の中から近傍の密度が最も高い点 (図 9(b) の赤色の星マーク) を選択する．ただし，推定の信頼性を向上させるため，質問画像の連結成分と参照画像の連結成分の面積比を R，アフィン変換行列から求められたアフィン変換行列の拡大率を β としたとき，$T_{\mathrm{area}} \leqq R/\beta^2 \leqq 1/T_{\mathrm{area}}$ を満たすもののみを推定に用いた．連結成分の仮の認識結果が正しければ $R/\beta^2 = 1$ になるため，この値が 1 から離れていれば結果が信頼できないことを意味する．本論文では $T_{\mathrm{area}} = 0.7$ とした．

最後に，連結成分毎に認識結果を定める．図 9(c) のような連結成分の回転角と連結成分番号が作る 2 次元空間での密度推定を利用して，尤もらしい回転角のパラメータと連結成分番号の組を推定する．推定には「候補グループ」に属する連結成分のアフィン変換行列を用いる．図 9(b) の場合と異なるのは，回転角は連続量であるが，連結成分番号は離散であるため，各連結成分番号について 1 次元の密度推定を行うことである．以上の処理により，各連結成分の種類 (連結成分番号) と姿勢 (せん断変形，独立変倍，回転角) を求めることができる．

5. 実　　験

提案手法の有効性を確認するため，2 つの認識実験を行った．実験では CPU が Opteron 2.6GHz で，メモリが 32GB である計算機を用いた．画像の登録と認識に要する計算量を削減するため，連結成分の幅と高さの大きい方が，参照画像では 100 ピクセル，質問画像では 50 ピクセルになるようにそれぞれ大きさを正規化した．

5.1 各種フォントに対する性能評価

提案手法の有効性を調べるために，図 10(a) に示す Arial，Century，Gigi，Impact の 4 種類のフォントの数字とアルファベット (各 62 字種) を認識した．ただし 3. 節で既に述べたように，一部の文字はアフィン歪みを受けると外見が類似してしまい，判別が困難なため，表 1 で示した文字は同一クラスとみなしている．また，4. 2. 4 節で述べた認識処理において，得票数が 0 だった場合はリジェクトとした．

(a) 上から Arial，Century，Gigi，Impact.　(b) ピクトグラム.

図 10　実験に用いた (a) フォントと (b) ピクトグラム.

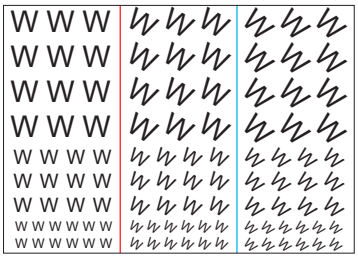


図 11　認識対象の紙面.

表 2　1 文字 (図形) の認識に要する平均処理時間．Picto はピクトグラムを表す.

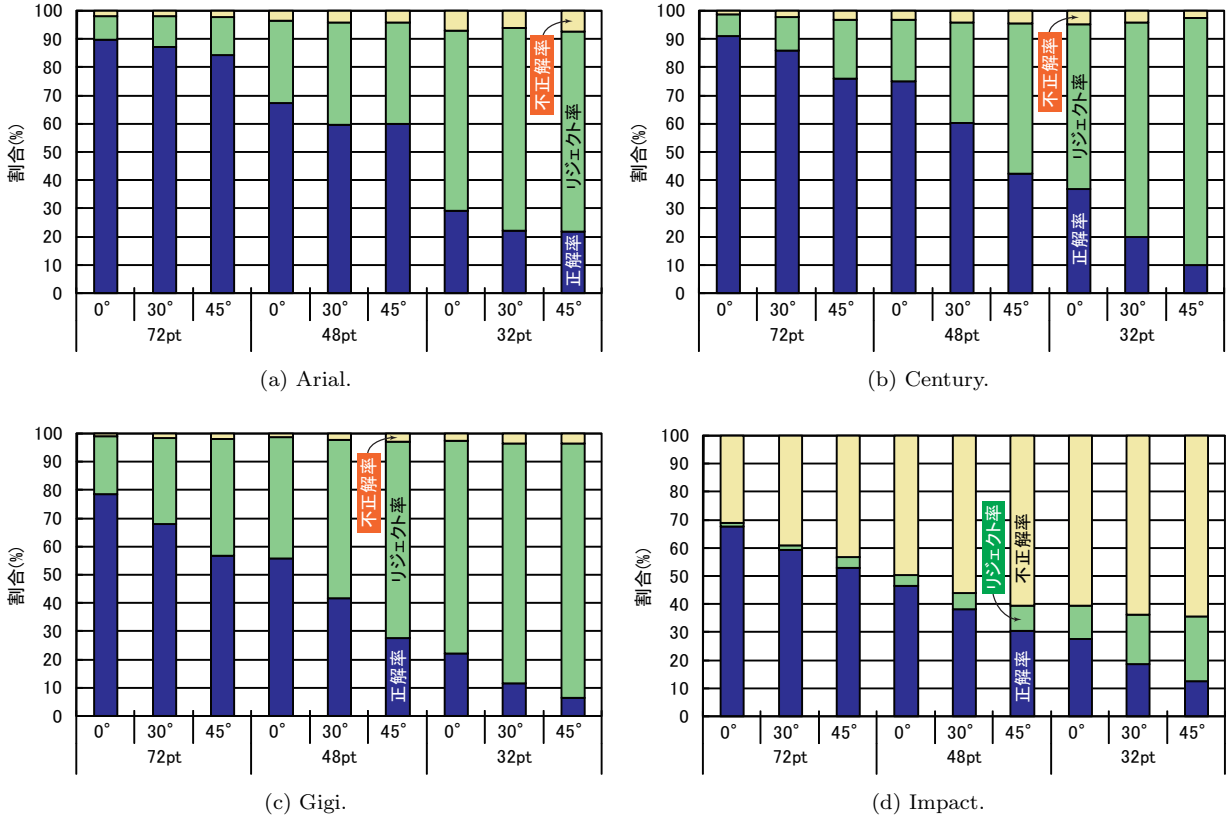
フォント	Arial	Century	Gigi	Impact	Picto
処理時間 (ms)	3.9	3.6	3.7	5.2	4.2

認識対象として，図 11 に示すテストパターンを作成した．この中には文字の大きさが 3 種類 (72pt，48pt，32pt)，文字の傾きが 3 種類 (0 度，30 度，45 度) の 9 種類の条件が含まれている．1 条件につき 12 文字なので，合計 108 文字が含まれている．これを 62 字種分，62 枚作成した．印刷したパターンは，デジタルカメラの紙面に対する傾きが 0 度，30 度，45 度の 3 種類になるように撮影した．72pt の Arial の「A」を正面 (0 度) から撮影したときの文字サイズの平均は 40.7×44.8 ピクセルであり，32pt の Arial の「A」を 45 度の角度から撮影したときは 10.0×18.6 ピクセルであった．実験では，認識対象のフォントのみを参照画像として登録した．4. 2. 3 節で述べたパラメータ S は 20 とした．

まず，1 文字当たりの平均処理時間を表 2 に示す．1 文字に要する処理時間は概ね 4ms であるので，単純計算で 1 秒間に 200〜250 文字程度の認識が可能と考えられる．

次に認識結果を図 12 に示す．図から，撮影角度の増加や文字の縮小に伴って正解率が減少していること，撮影角度よりも文字の縮小による影響のほうが大きいことがわかる．認識結果について，以下で詳しく考察する．

まず Impact 以外のフォントでは，正解率が減少すると，その分リジェクト率が増加し，誤認識率はあまり増えなかった．この原因は，特徴ベクトルの量子化パラメータ D が大きかったためと考えられる．より詳しく言うと，D が大きいと，画像のわずかな変動でも計算されるハッシュのインデックスが変わるため，信頼できる仮の認識結果が得られなくなる．ここで D を 3 から 2 に



(a) Arial.　(b) Century.

(c) Gigi.　(d) Impact.

図 12　文字の大きさと撮影角度を変化させたときの認識率.

変更すれば，Arial で最も認識率の低かった「32pt の文字を 45 度から撮影した場合」でも，正解率が 21.54%から 52.73%に上昇する．しかし，不正解率も 7.57%から 36.31%に上昇してしまう．どちらを選ぶかは応用次第であるが，一般に誤認識が少ないことは重要な性質と考えられるため，本論文では $D = 3$ を採用した．

次に Impact では，正解率が減少してもほとんどリジェクトされず，正解率の減少分はほとんど誤認識率になった．この原因としては，Impact は線幅が太いため，どの連結成分の特徴ベクトルも似てしまい，十分な識別性能を得られなかったことが考えられる．表 2 の平均処理時間を見ると，他のフォントより処理に時間がかかっている．これは特徴ベクトルの識別性能が足りず，ハッシュの衝突が多数起きていることを示唆している．この問題については，2.2.2 節で述べたように，既存の正規化手法や特徴量の導入で改善できると考えている．

最後に，文字以外の図形の認識可能性を調べるため，上記の 4 フォントの他に図 10(b) に示す 10 種のピクトグラムも同様の方法で認識した．図 13 と表 2 に示すように，Impact 以外のフォントと同様の結果が得られた．

以上により，提案手法が高速で動作し，一部のフォントを除けば誤認識率が少ないことが確認できた．

5.2　図 1 の文書に対する性能評価

図 1 に示した文書を認識した．デジタルカメラを紙面から 0 度，30 度，45 度の 3 種類に傾けて撮影して，背景が写らないように紙面の部分だけ切り取った．切り取った後の 0 度，30 度，45 度の画像の大きさはそれぞれ 2470×1746，2278×1746，2038×1844 である．得られた画像を図 14 に示す．図 1 には 236 文字含まれている (カンマは除く)．内訳は，Arial が 168 文字，Arial Black Italic が 27 文字，MIRU のロゴの周囲の文字 (フォント不明) が 41 文字である．このうち，Arial と MIRU のロゴの周囲の文字を登録して認識実験を行った．

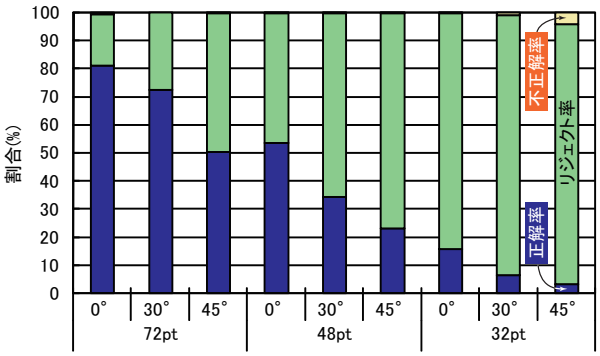


図 13　ピクトグラムの大きさと撮影角度を変化させたときの認識率.

$S = 200$ と $S = 20$ について，処理時間と認識結果を表 3 に示す．主な誤認識は「i」の認識失敗 (「i」と「I」や「l」の混同),「U」と「u」や「n」の混同,「E」と「m」の混同であった．$S = 200$ と $S = 20$ を比べると，$S = 200$ は $S = 20$ よりも認識率が高かったが，$S = 20$

(a) 0 度

(b) 30 度

(c) 45 度

図 14　図 1 の画像をデジタルカメラで撮像して得た認識対象.

表 3　図 1 を認識したときの処理時間と認識結果.

試行回数 S	200			20		
角度 (度)	0	30	45	0	30	45
処理時間 (ms)	7990	7990	7020	1300	1260	1140
認識率 (%)	94.9	90.7	86.4	86.9	81.8	76.3
リジェクト率 (%)	0.4	3.0	6.4	6.4	9.3	16.5
誤認識率 (%)	4.7	6.4	7.2	6.8	8.9	7.2

の処理時間は $S = 200$ の約 1/6 であった．以上より，提案手法は S が小さい場合は高速な認識が可能であること，若干処理時間を要するが，必要であれば S を大きくすることで，より頑健な認識が実現できることが確認できた．

6. 関 連 研 究

1. 節で述べた手法以外に関連する研究をまとめておく．

提案手法のようにアフィン変換を受けた連結成分を認識する方法としては，アフィン変換を受けた図形そのものを正規化し，その後に照合する方法が知られている．Leu らは図形を構成する画素の共分散行列を求め，その逆行列を用いることで拡大・縮小，せん断変形を正規化する方法を提案している [8]．しかし，この方法では回転に任意性を残すため，何らかの方法で図形の回転を考慮した照合が必要である．堀松らは Leu の方法で正規化した図形の高速な照合法を検討している [9] が，実時間の処理は困難である．

この問題に対して，提案手法のアフィン変換の方法を用いるか，提案手法を相似変換レベルで用いれば，どちらの場合も $O(P^2)$ の計算量で照合可能である．ただし，提案手法の計算量は Leu の正規化法を事前に適用しなくても全く変わらないため，結局は提案手法を単独で用いたほうが高速になる．

また，中居らは，GH の高速化手法と捉えることもできる Locally Likely Arrangement Hashing(LLAH) という手法を提案している [10]．この手法は，特徴点の局所的な配置に着目し，幾何学的不変量とハッシュを用いて，データベースからクエリに対応する特徴点を高速に検索する．LLAH の計算量とメモリ使用量は，条件によっては GH の数億分の一で済むことが知られている．この性能向上の鍵は，特徴点の組み合わせを限定して計算量を減少させていることと，特徴ベクトルの高次元化による識別性能の向上である [5]．前者は提案手法の考え方と類似しているが，LLAH では特徴点が分散していることを想定して不変量を計算するため，本論文で扱う問題のように，特徴点が連結している場合には適用が困難である．後者については提案手法にも適用することができ，提案手法の更なる性能向上につながる可能性がある．

7. む す び

本論文では，カメラを用いた文字認識において，(1) 実時間処理が可能である，(2) 射影歪みに頑健である，(3) 文字の配置に依らない認識が可能である，という 3 要件を同時に満たす文字認識手法を提案した．提案手法は，添付資料中の動画で示すように，web カメラと接続したノートパソコン上で実時間で動作することができる．

今後の課題としては，本文中に記したアイディア (文字認識に使用される既存の正規化手法や特徴量の導入，LLAH のような特徴ベクトルの高次元化等) の検討が挙げられる．

謝辞　本研究の一部は，科研費補助金若手研究 (B)(19700177，21700202) の補助による．

文　　献

[1] Y. Watanabe, Y. Okada, Y.-B. Kim and T. Takeda, “Translation camera, ” Proc. ICPR1998, pp.613–617, 1998.

[2] G. K. Myers, R. C. Bolles, Q.-T. Luong, J. A. Herson and H. B. Aradhye, “Rectification and recognition of text in 3-d scenes,” IJDAR, vol.7, no.2-3, pp.147–158, 2004.

[3] Y. Kusachi, A. Suzuki, N. Ito and K. Arakawa, “Kanji recognition in scene images without detection of text fields—robust against variation of viewpoint, contrast, and background texture—, ” Proc. ICPR2004, 2004.

[4] L. Li and C. L. Tan, “Character recognition under severe perspective distortion, ” Proc. ICPR2008, 2008.

[5] 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, ” Vol. 2, pp.1010–1019, Sept. 2007.

[6] H. Ishida, S. Yanadume, T. Takahashi, I. Ide, Y. Mekada and H. Murase, “Recognition of low-resolution characters by a generative learning method, ” Proc. CBDAR2005, pp.45–51, 2005.

[7] M. Iwamura, R. Niwa, A. Horimatsu, K. Kise, S. Uchida and S. Omachi, “Layout-free dewarping of planar document images, Proc. DRR XVI, ” 7247-36, Jan. 2009.

[8] J.-G. Leu, “Shape normalization through compacting,” Pattern Recognition Letters, vol.10, no.4, pp.243–250, 1989.

[9] A. Horimatsu, R. Niwa, M. Iwamura, K. KIse, S. Uchida and S. Omachi, “Affine invariant recognition of characters by progressive pruning, ” Proc. DAS2008, pp.237–244, Sept. 2008.

[10] 中居友弘，黄瀬浩一，岩村雅一，“特徴点の局所的配置に基づくデジタルカメラを用いた高速文書画像検索，” 電子情報通信学会論文誌 D，vol.J89-D, no.9，pp.2045–2054，Sept. 2006.