

## 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

中居 友弘<sup>†</sup>      黄瀬 浩一<sup>†</sup>      岩村 雅一<sup>†</sup>

Fast Camera-Based Document Image Retrieval Using Local Arrangements of  
Feature Points

Tomohiro NAKAI<sup>†</sup>, Koichi KISE<sup>†</sup>, and Masakazu IWAMURA<sup>†</sup>

あらまし 本論文では、デジタルカメラで撮影して得られた文書画像をもとに、データベースから対応する文書画像を見つける文書画像検索の手法を提案する。従来のスキャナを利用した文書画像検索と異なり、デジタルカメラを利用する場合には射影ひずみなどのデジタルカメラ特有の問題に対処する必要がある。また、大規模データベースでの利用を可能にするため、検索の効率性も重要である。これらの問題に対処するため、本論文では特徴点の局所的配置を特徴量とし、ハッシュ表を用いた投票によって検索する手法を提案する。本手法では、射影ひずみを実用的な範囲に制限し、特徴点の組合せを局所領域に限定することで、計算量の削減を実現する。また、高精度な検索を実現するため、射影変換の不変量である複比で特徴点の配置を表現し、特徴量としている。実験により、10,000 ページのデータベースから精度 98%、平均処理時間約 160 ms (特徴点抽出の画像処理に要する約 1s を除く) で検索できることが示された。

キーワード 文書画像検索、デジタルカメラ、複比、ハッシュ、投票

### 1. ま え が き

文書画像検索とは、データベースから利用者の与えた検索質問に対応する文書画像を検索する処理である。文書画像検索の検索質問には、キーワード、レイアウト、撮影された画像など、利用目的に応じて様々なものがある [1]。本論文では、撮影した文書画像を検索質問としてデータベースから同一の文書に対する文書画像を検索する場合を考える。このような文書画像検索には、応用として様々なサービスが考えられる。例えば、印刷文書に書込みがなされていた場合、対応する文書画像との差分をとることで書込み抽出が可能になる。また、データベースの文書画像にあらかじめ関連情報を定義しておくことで、検索により関連情報の取得が可能になる。

このような文書画像検索に関する従来手法として、入力機器にスキャナを用いる手法 [2], [3] が提案されて

いる。これらの手法では、得られた文書画像を行ごとくに分離し、各行における文字の並びを特徴量として文書画像を検索する。しかし、スキャナを用いる場合、画像の取込みに要する手間が大きく、持ち運びも難しいため、利便性に問題がある。

本研究では、文書画像検索の入力機器としてより利便性の高いデジタルカメラを用いることを考える。デジタルカメラを用いる場合、機器の利便性が高い反面、射影ひずみや不均一な照明、フォーカスのぼけなどのカメラ特有の問題が生じる [4] ~ [6]。そのため、従来手法をそのまま適用することはできない。

デジタルカメラで撮影された文書画像における射影ひずみの問題については、射影変換パラメータを推定して射影ひずみを補正する手法 [7] が検討されている。射影ひずみを補正し、正対した文書画像が得られれば、スキャナを利用する従来手法を適用することができる。しかし、射影ひずみの補正には文書画像上の行端の位置などの情報が必要であり、これらの情報は撮影条件によっては必ずしも得られるとは限らないという問題点がある。

<sup>†</sup> 大阪府立大学大学院工学研究科，堺市  
Graduate School of Engineering, Osaka Prefecture University,  
1-1 Gakuen-cho, Naka-ku, Sakai-shi, 599-8531 Japan

ところで、コンピュータビジョンの分野では、画像のひずみを補正することなく、ひずみに不変な形で登録及び検索を行う手法として、Geometric Hashing (GH)[8]が知られている。GHでは、画像から特徴点を抽出し、特徴点に基づいて変換によらない形で登録及び検索を行う。しかし、登録及び検索の際に画像全体の特徴点の組合せを用いるため、特徴点の数が多くなると膨大な計算量が必要となる。具体的には、画像の特徴点の数を  $N$  としたとき、射影ひずみに不変なものにするためには  $O(N^5)$  の計算量が必要となる。このように計算量が大きいと、文書画像のように多数の特徴点をもつ画像の検索に用いることは困難となる。

そこで本論文では、特徴点の多い文書画像に対しても高速な検索が実現できる、 $O(N)$  の文書画像検索法を提案する。提案手法では、特徴点の局所的な配置を用いて特徴量を求め、これを用いて特徴点をインデクシングする。特徴点の近傍構造は射影ひずみによって変化する可能性があるため、様々な可能性を記録しておく。具体的には、近傍  $n$  点のうち  $m$  点は保存されるという前提のもと、可能な  $m$  点の組合せをすべて求め、その配置から得られる特徴量を特徴点のインデックスとする。局所的な配置の数は特徴点全体の数に対して定数であるため、 $O(N)$  の計算量を実現できる。特徴点の配置については、射影変換の不変量である複比を用い、 $m$  点から計算される複比の列という形で記述する。また、インデクシングされた特徴点は、ハッシュ表に格納することによって、高速な検索を実現する。

実験の結果、提案手法では紙面に対して  $45^\circ$  程度の角度から撮影された検索質問でも検索が可能であり、 $60^\circ$  程度の角度から撮影された検索質問の場合、10,000 ページのデータベースでは精度が 98%、検索質問当りの平均処理時間が約 160 ms (特徴点抽出の画像処理に要する約 1s を除く) であることが確認された。

## 2. 提案手法

### 2.1 処理の流れ

提案する文書画像検索システムの構成を図 1 に示す。登録文書 (registered document) 及び検索質問 (query image) は、特徴点抽出 (feature point extraction) により点の集合に変換された後、登録の場合は登録処理 (registration) へ、検索の場合は検索処理 (retrieval) に入力される。これらの処理では、

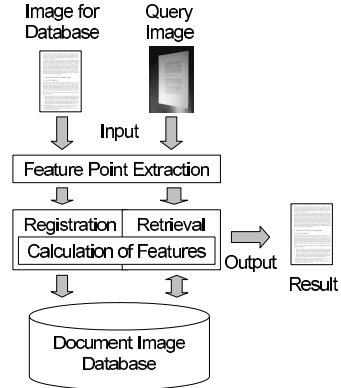


図 1 文書画像検索システム

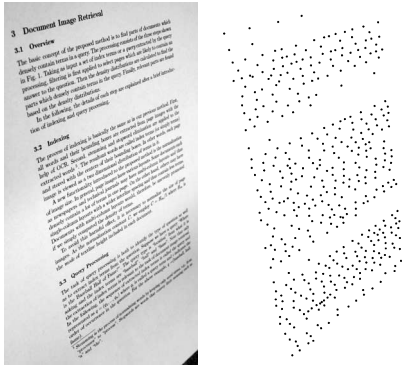
Fig. 1 Document image retrieval system.

特徴点から複比を用いた特徴量を求める特徴量計算 (calculation of features) に同じ処理を用いる。登録処理では、特徴点から得られた特徴量をハッシュ表のインデックスに変換し、これを用いて文書画像データベース (document image database) へ特徴点及び文書の識別番号を登録する。一方、検索処理では、特徴量から同様にインデックスを計算してデータベースから対応する特徴点を検索する。そして、特徴点の属する文書の識別番号に対して投票することで所望の文書画像を検索する。このように、提案手法では特徴点の検索を通じて文書の検索を行う。特徴点を用いることにより、ノイズや隠れなどの影響で文書画像が部分的に変化した場合でも正しい検索結果を得ることが期待できる。以下、各々について述べる。

### 2.2 特徴点抽出

特徴点抽出では、複比の計算に用いる特徴点を求める。ここで重要なことは、特徴点の再現性、すなわち射影ひずみ・ノイズ・低解像度の影響下でも同一の特徴点を得られることである。英文文書における単語の重心は、この条件を満たす特徴点の一つである。これは、英文文書では単語と単語の間に空白があり、分離が比較的容易なためである。

処理の概要は以下のとおりである。まず、入力画像を適応 2 値化 [9] により 2 値画像に変換する。次に、2 値画像に対してガウスフィルタを適用し、再度 2 値化することで単語領域を得る。ここで、ガウスフィルタのパラメータは連結成分の面積の最頻値によって求められる文字サイズの推定値に基づいて適応的に定める。最後に、各単語領域の重心を求め、これを特徴点とする。図 2(a) に示す入力画像からは、図 2(b) のよう



(a) Input image. (b) Feature points.

図 2 特徴点抽出

Fig. 2 Feature point extraction.

な特徴点が抽出される。

### 2.3 複比

次に、特徴量の計算に用いる複比について述べる。複比は、射影変換の不変量であり、同一平面上の 5 点 ABCDE の座標から以下の式で計算される [10]。

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

ここで、 $P(A, B, C)$  は頂点  $A, B, C$  がなす三角形の面積である。複比は射影変換の不変量であるため、点 ABCDE の座標が射影ひずみによって変化しても、その値は同一に保たれるという性質をもつ。

さて、画像上の特徴点から得られる複比の値は連続値だが、インデックスに用いる際には離散化する必要がある。本手法では、予備実験により特徴点から得られる複比のヒストグラムを作成し、各離散値における複比の頻度が等しくなるように離散値を割り当てる。

### 2.4 特徴量計算

#### 2.4.1 特徴量の満たすべき条件

提案手法における特徴量とは、文書画像の特徴点を表現する量である。検索質問及び登録文書のそれぞれについて、特徴点から得られる特徴量を計算し、それらの値を比較することで検索質問と登録文書が対応しているか否かを判断する。

高精度な検索を実現するためには、特徴量は以下に示す二つの条件を満たす必要がある。第一の条件は、各種ひずみの影響を受けても同じ文書の同じ特徴点からは同じ特徴量が得られなければならないということである。もし登録文書と検索質問から異なる特徴量が得られれば、検索によって正しく対応する特徴点を見

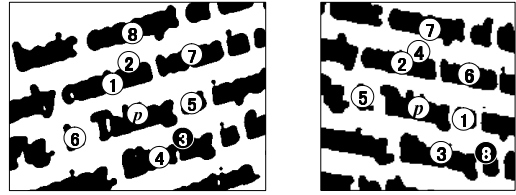


図 3 射影ひずみによる近傍点の変化 (数字は点  $p$  との距離の順位)

Fig. 3 Change of nearest feature points by perspective distortion. Numbers indicate ranks of distances from the point  $p$ .

つけることはできない。この条件を「特徴量の安定性」と呼ぶ。第二の条件とは、異なる特徴点からは異なる特徴量が得られなければならないということである。もし異なる特徴点から同じ特徴量が得られれば、検索の際に正しく対応する特徴点だけでなく対応しない特徴点まで見つかることになる。この条件を「特徴量の識別性」と呼ぶ。

また、いうまでもなく、安定性や識別性の高い特徴量であっても、計算量が膨大であれば利用は困難となる。したがって、特徴量は精度に関する上記の 2 条件を満たすだけでなく、計算量も小さい必要がある。

これら三つの条件が高精度な文書画像検索における特徴量の満たすべき条件である。

#### 2.4.2 特徴量の安定性

まず特徴量の安定性について述べる。前述したように、提案手法では各特徴点の近傍点から複比を用いて特徴量を計算する。そのため、特徴量が安定であるためには、射影ひずみによって近傍点の座標が変化しても特徴量の計算に用いる特徴点に同じ特徴点が必要がある。図 3 に示されるように、近傍点は射影ひずみの影響で変化するため、特徴点  $p$  の近傍 5 点から計算される複比を特徴量とすると、同じ特徴点  $p$  から同じ特徴量を得ることはできない。

そこで、提案手法ではより広い範囲の近傍点から複数の特徴点の組合せを作成し、そこから複数の特徴量を計算する。これは、図 3 のように、射影ひずみの影響があってもある程度広い範囲の近傍  $n$  点 (図 3 では 8 点) のうち  $m$  点 (図 3 では 7 点) までは同じ点が含まれるという仮定に基づいている。近傍  $n$  のうち  $m$  点在同一であるならば、図 4 のように  $n$  点からすべての  $m$  点の組合せ  $P_{m(0)}, P_{m(1)}, \dots, P_{m(nC_m-1)}$  を作成し、特徴量を計算することで、少なくとも一つは同じ特徴量が得られると考えられる。

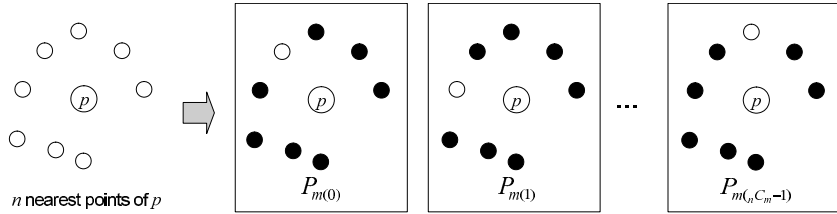


図 4  $n (= 8)$  点からの  $m (= 7)$  点のすべての組合せ  
 Fig. 4 All possible combinations of  $m (= 7)$  points from  $n (= 8)$  nearest points.

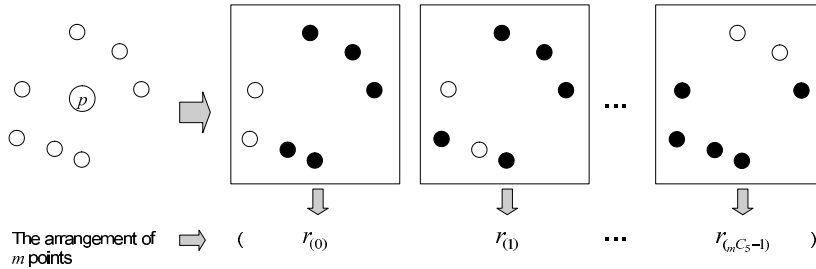


図 5 すべての 5 点の組合せから得られる複比で表現される  $m (= 7)$  点の配置  
 Fig. 5 The arrangement of  $m (= 7)$  points described as a sequence of cross-ratios calculated from all possible combinations of 5 points.

2. 4. 3 特徴量の識別性

次に、特徴量の識別性について述べる。提案手法では、一つの特徴量の計算に用いる特徴点の数  $m$  を増やすことで識別性を高める。

$m$  点の配置の表現方法としては、図 5 のように  $m$  点から得られるすべての 5 点の組合せから計算される複比の列  $r_{(0)}, r_{(1)}, \dots, r_{(m, C_m-1)}$  を用いる。  $m$  が大きければ大きいほど、そこから計算される複比の数が多くなるため、対応すべきでない特徴点から同じ特徴量が偶然に現れる可能性は低くなる。

ただし、  $m$  が大きすぎると特徴量の安定性が低下する。なぜなら、特徴量が一致するためには特徴量の複比すべてが一致する必要があるが、  $m$  が大きいと特徴量の複比の数  $m C_m$  が増加するので、誤差の影響で異なる複比が計算される可能性が高くなるためである。

2. 4. 4 計算量と記憶容量

以上のように、提案手法では特徴量の安定性を高めるために広い範囲の近傍  $n$  点から複数の特徴量を計算し、識別性を高めるために一つの特徴量の計算に用いる特徴点の数  $m$  を増やしている。しかし、これらのパラメータを大きくしすぎると計算量の面で問題が生じる。  $n$  及び  $m$  が大きすぎると、複比の計算回数が増加する。そのため、登録及び検索に要する処理時間が大きくなる。また、登録時には計算される特徴量を記

憶しておく必要があるため、大きな記憶容量が必要となる。

2. 4. 5 複比の量子化レベル

安定性及び識別性に影響するパラメータは  $n, m$  だけではない。複比の量子化レベル  $k$  もこれらに影響する。  $k$  の値が大きいく (複比が細かく離散化される) と、誤差の影響で同じ 5 点から計算される複比が異なる値に離散化される可能性が高くなり、安定性が低下する。  $k$  の値が小さい (複比が粗く離散化される) と、異なる 5 点から計算された複比が同じ値に離散化される可能性が高くなり、識別性が低下する。

以上のことから、高い検索精度と同時に高速な処理や小さい記憶容量を実現するには、  $n, m$  及び  $k$  を適切に設定する必要があるといえる。

2. 5 登 録

以上の準備に基づき、図 6 に示す登録アルゴリズムについて説明する。登録処理では、登録文書の各特徴点について、その近傍  $n$  点からすべての  $m$  点の組合せを生成し、そこから複比を用いたインデックスを求めて図 7 に示すハッシュ表に登録する。

以下、アルゴリズムに沿って説明する。1 行目で特徴点の集合から 1 点を取り出して  $p$  とし、2 行目で  $p$  の近傍  $n$  点を取り出して  $P_n$  とする。3 行目で  $P_n$  から  $m$  点を取り出して  $P_m$  とし、4 行目で

```

1: for each  $p \in \{ \text{特徴点} \}$  do
2:    $P_n \leftarrow p$  の近傍  $n$  点の集合
3:   for each  $P_m \in \{ P_n \text{ から } m \text{ 点を選ぶすべての組み合わせ} \}$  do
4:      $P_m$  から適当に 1 点  $p_0$  を選び,  $P_m$  の要素を  $p$  を中心として  $p_0$  を先頭に時計回りに並べて特徴点の列  $L_m = (p_0, \dots, p_{m-1})$  を作る. 以後,  $p_i$  の順序とは添字の若い順のことをいう
5:      $(L_{5(0)}, \dots, L_{5(i)}, \dots, L_{5(mC_5-1)}) \leftarrow L_m$  から 5 点を選んでできる部分列  $L_5$  をすべて求めて辞書式順序に並べたリスト
6:     for  $i = 0$  to  $mC_5 - 1$  do
7:        $r_{(i)} \leftarrow L_{5(i)}$  から計算される複比の離散値
8:     end for
9:      $H_{\text{index}} \leftarrow$  式 (2) で計算されるハッシュ表のインデックス
10:     $H_{\text{index}}$  を用いて文書 ID, 点 ID,  $r_{(0)}, \dots, r_{(mC_5-1)}$  をハッシュ表に登録
11:  end for
12: end for

```

図 6 登録アルゴリズム  
Fig. 6 Registration algorithm.

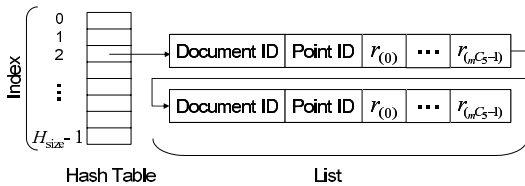


図 7 ハッシュ表の構造  
Fig. 7 Structure of the hash table.

$P_m$  の要素に対して  $p$  を中心とした時計回りの順序を与えて特徴点の列  $L_m$  を定める. 次に, 5 行目で  $L_m$  の要素から順序を保ったまま 5 点を選択してできる特徴点の列  $L_{5(i)}$  を求め, 辞書式順序に並べる. 例えば,  $m = 7$  のときには,  $(L_{5(0)}, \dots, L_{5(7C_5-1)}) = ((p_0, p_1, p_2, p_3, p_4), (p_0, p_1, p_2, p_3, p_5), \dots, (p_2, p_3, p_4, p_5, p_6))$  が得られる. 更に 7 行目において,  $L_{5(i)}$  の点を順に式 (1) における  $A, B, \dots, E$  として複比を求め, 離散化して  $r_{(i)}$  とする. このようにして得られた複比  $r_{(i)}$  より, 9 行目で式 (2) に示されるハッシュ関数を用いてハッシュ表のインデックス  $H_{\text{index}}$  を求め, 10 行目で  $H_{\text{index}}$  を用いて登録文書の識別番号である文書 ID (Document ID) と特徴点の識別番号である点 ID (Point ID), 複比  $r_{(i)} (i = 0, 1, \dots, mC_5 - 1)$  をハッシュ表に登録する.

本論文で用いるハッシュ関数を以下に示す.

$$H_{\text{index}} = \left( \sum_{i=0}^{mC_5-1} r_{(i)} k^i \right) \bmod H_{\text{size}} \quad (2)$$

ここで,  $k$  は複比の量子化レベル,  $H_{\text{size}}$  はハッシュ表

```

1: for each  $p \in \{ \text{特徴点} \}$  do
2:    $P_n \leftarrow p$  の近傍  $n$  点の集合
3:   for each  $P_m \in \{ P_n \text{ から } m \text{ 点を選ぶすべての組み合わせ} \}$  do
4:     for each  $p_0 \in P_m$  do
5:        $P_m$  の要素を  $p$  を中心として  $p_0$  を先頭に時計回りに並べて特徴点の列  $L_m = (p_0, \dots, p_{m-1})$  を作る
6:        $(L_{5(0)}, \dots, L_{5(i)}, \dots, L_{5(mC_5-1)}) \leftarrow L_m$  から 5 点を選んでできる部分列  $L_5$  をすべて求めて辞書式順序に並べたリスト
7:       for  $i = 0$  to  $mC_5 - 1$  do
8:          $r_{(i)} \leftarrow L_{5(i)}$  から計算される複比の離散値
9:       end for
10:       $H_{\text{index}} \leftarrow$  式 (2) で計算されるハッシュ表のインデックス
11:       $H_{\text{index}}$  を用いてハッシュ表にアクセスし, リストを得る
12:      for each リストの各要素について do
13:        if 条件 1~3 が成立する then
14:          投票テーブルの文書 ID に投票
15:        end if
16:      end for
17:    end for
18:  end for
19: end for
20: 得票数から式 (3) を用いて各文書のスコアを得る
21: 最大のスコアをもつ文書を検索結果とする

```

図 8 検索アルゴリズム  
Fig. 8 Retrieval algorithm.

のサイズである. また, 登録時に衝突が生じた場合, データは図 7 のようにリスト構造で付け加えられる.

このような処理をすべての  $p$  について施し, 文書を登録する.

## 2.6 検索

次に検索について述べる. 図 8 に検索アルゴリズムを示す. 登録処理と同様に, 1~3 行目で,  $p, P_n, P_m$  を得る. 4~5 行目では, 登録処理とは異なり,  $P_m$  のすべての点を開始点  $p_0$  として  $L_m$  を作成する. これは, 図 6 の登録アルゴリズムの 4 行目において, 画像の回転を考慮せずに,  $P_m$  から得られる列として  $L_m$  を一つだけ登録しているために必要となる. 点  $p$  を中心とした特徴点の順序は, 射影変換を受けた後でも, 開始点の任意性を残して一定である. すなわち,  $L_m$  の巡回置換を考えると, その中に必ず一つは登録時に用いた順序のものが存在するといえる. 6~10 行目で登録時と同様の手法によりハッシュ表のインデックスを求め, 11 行目でハッシュ表を読んで図 7 にあるようなリストを得る. 12~14 行目でリストの各要素について, 文書 ID の一次元配列である投票テーブルの該当するセルに投票する.

ただし,  $H_{\text{index}}$  でハッシュ表に登録されている要素

の複比の列  $r_{(0)}, \dots, r_{(m, C_5-1)}$  は、ハッシュ関数で変換されているため、同一の  $H_{index}$  であっても必ずしも検索質問のものとは一致するとは限らない。一致しない場合を除くために以下の条件が用いられる。

条件 1:  $r_{(0)}, \dots, r_{(m, C_5-1)}$  が 7~9 行で計算されるものと完全に一致する

また、図 9 に示されるような誤った対応による投票も問題となる。提案手法では、以下の 2 種類の誤対応について対処する。第一は、検索質問の一つの特徴点 (図 9 では A) が登録文書の二つ以上の特徴点 (図 9 では A' 及び A'') と対応するものであり、第二は、検索質問の二つ以上の特徴点 (図 9 では A 及び B) が登録文書の一つの特徴点 (図 9 では A') と対応するものである。これらの誤対応による投票を制限するため、以下の条件を考える。

条件 2: 検索質問の特徴点  $p$  から同一の登録文書に投票されていない

条件 3: 検索質問の特徴点のうち、特徴点  $p$  とは別の特徴点から、同一登録文書の同一点に投票されていない

このような処理をすべての特徴点について行くと、各登録文書の得票数が投票テーブルに得られる。しかし、このようにして得られた得票数は依然として誤った対応によるものを含んでいる。誤った投票は登録文書に含まれる特徴点数におおむね比例するため、多くの特徴点をもつ登録文書は不当に多くの得票数を得ることになる。このような誤った投票を補正するため、以下の式のように文書  $d_i$  に対してスコア  $S(d_i)$  を定義する。

$$S(d_i) = V(d_i) - c \cdot N(d_i) \tag{3}$$

ここで、 $V(d_i)$  は  $d_i$  の得票数、 $N(d_i)$  は  $d_i$  に含まれる特徴点数、 $c$  は予備実験で定められる特徴点数と誤投票の比例定数である。最終的に、最大のスコアを

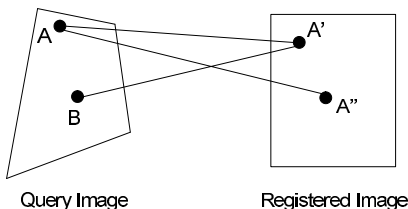


図 9 誤った点対応  
Fig. 9 Erroneous correspondences between feature points.

得た文書を検索結果とする。

### 3. 実験

#### 3.1 実験概要

提案手法の有効性を検証するために、図 10 に示すような英語論文の PDF ファイルを変換して得られた文書画像のデータベースと、印刷文書をデジタルカメラで撮影した検索質問を用いて実験を行った。文書画像データベースとしては、それぞれ 10, 100, 1,000, 10,000 ページの文書画像からなるデータベース A, B, C, D を用いた。データベース D は C を、C は B を、B は A をその一部として含んでいる。また、PDF ファイルとしては主に CVPR, ICPR, ICCV などの類似したレイアウトをもつ国際学会の予稿集のものを用いた。検索質問としては、データベース B を印刷したもの紙面に対してそれぞれ  $90^\circ, 60^\circ, 45^\circ, 30^\circ$  程度の角度から撮影した検索質問 1, 2, 3, 4 を用いた。画像の数はそれぞれ 100 枚である。検索質問の例を図 11 に示す。検索質問の作成には、CANON EOS Kiss Digital (630 万画素) と付属のレンズ EF-S 18~55 mm USM を使用した。実験に用いたハッシュ表のサイズは  $H_{size} = 2^{27} - 1$  とした。また、CPU が AMD Opteron 1.8 GHz, メモリが 6 GByte の計算機を用いた。

#### 3.2 実験 1: パラメータ $n, m$ と性能の関係

提案手法の性能は、特徴量計算時の組合せ数を決めるパラメータ  $n$  及び  $m$  の値によって変化する。そこでまず、様々な  $n, m$  の組合せにおける検索精度、処理時間、記憶容量について調べた。実験にはデータベース B 及び検索質問 1~4 を用いた。

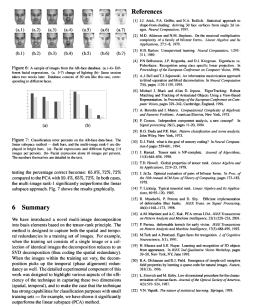


図 10 データベース中の登録文書の例  
Fig. 10 Examples of registered images in the database.

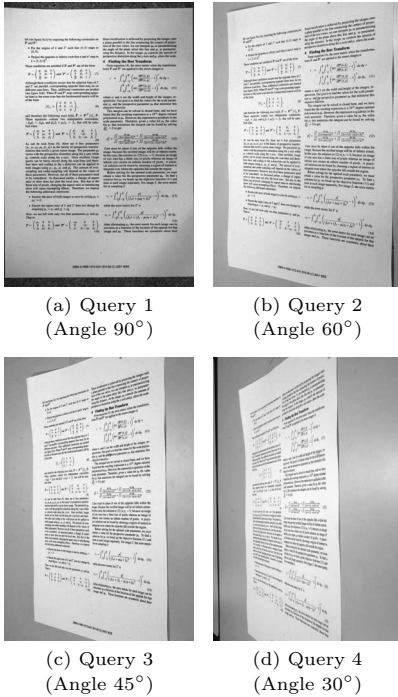


図 11 検索質問の例  
Fig. 11 Examples of query images.

表 1 パラメータと性能の関係 (撮影角度 90°)  
Table 1 The relation between parameters and performance (angle 90°).

$n$	$m$	${}_n C_m$	${}_m C_5$	$k$	精度 [%]	処理時間 [msec.]	リスト数	$l$
6	6	1	6	30	100	18.5	$6.7 \times 10^4$	1.37
7	6	7	6	50	99	48.8	$4.7 \times 10^5$	1.48
	7	1	21	5	100	28.7	$6.7 \times 10^4$	1.15
8	6	28	6	140	86	136.0	$1.9 \times 10^6$	1.13
	7	8	21	12	100	122.4	$5.3 \times 10^5$	1.02
	8	1	56	8	99	52.8	$6.7 \times 10^4$	1.00
9	6	84	6	190	25	392.3	$5.6 \times 10^6$	1.18
	7	36	21	25	100	502.6	$2.4 \times 10^6$	1.02
	8	9	56	8	99	364.9	$6.0 \times 10^5$	1.00
	9	1	126	4	97	111.0	$6.7 \times 10^4$	1.00
10	6	210	6	110	15	1493.1	$1.4 \times 10^7$	2.04
	7	120	21	20	100	1661.7	$8.0 \times 10^6$	1.05
	8	45	56	5	100	1782.8	$3.0 \times 10^6$	1.02
	9	10	126	2	100	990.0	$6.7 \times 10^5$	1.01
	10	1	252	2	99	227.3	$6.7 \times 10^4$	1.00

検索質問の撮影角度ごとの実験結果を表 1~表 4 に示す。ここで、精度は検索質問に正しく対応する文書が最も多くの得票数を得た割合、処理時間は特徴点抽出処理を除く検索処理に要した時間である。リスト数は、ハッシュ表に登録される要素 (図 7 に示す、文書

表 2 パラメータと性能の関係 (撮影角度 60°)  
Table 2 The relation between parameters and performance (angle 60°).

$n$	$m$	${}_n C_m$	${}_m C_5$	$k$	精度 [%]	処理時間 [msec.]	リスト数	$l$
6	6	1	6	30	90	18.7	$6.7 \times 10^4$	1.38
7	6	7	6	40	79	53.4	$4.7 \times 10^5$	1.82
	7	1	21	5	100	28.3	$6.7 \times 10^4$	1.15
8	6	28	6	150	53	133.2	$1.9 \times 10^6$	1.11
	7	8	21	12	100	122.9	$5.3 \times 10^5$	1.02
	8	1	56	5	99	53.3	$6.7 \times 10^4$	1.00
9	6	84	6	240	10	372.0	$5.6 \times 10^6$	1.11
	7	36	21	24	100	502.2	$2.4 \times 10^6$	1.02
	8	9	56	8	99	366.0	$6.0 \times 10^5$	1.00
10	9	1	126	5	74	111.4	$6.7 \times 10^4$	1.00
	6	210	6	120	5	1408.3	$1.4 \times 10^7$	1.86
	7	120	21	21	100	1662.1	$8.0 \times 10^6$	1.05
	8	45	56	6	100	1784.7	$3.0 \times 10^6$	1.02
	9	10	126	2	100	994.9	$6.7 \times 10^5$	1.01
10	1	252	2	98	229.0	$6.7 \times 10^4$	1.00	

表 3 パラメータと性能の関係 (撮影角度 45°)  
Table 3 The relation between parameters and performance (angle 45°).

$n$	$m$	${}_n C_m$	${}_m C_5$	$k$	精度 [%]	処理時間 [msec.]	リスト数	$l$
6	6	1	6	30	28	18.8	$6.7 \times 10^4$	1.38
7	6	7	6	40	12	53.8	$4.7 \times 10^5$	1.82
	7	1	21	8	80	27.9	$6.7 \times 10^4$	1.03
8	6	28	6	180	8	131.7	$1.9 \times 10^6$	1.08
	7	8	21	14	99	123.8	$5.3 \times 10^5$	1.01
	8	1	56	5	70	53.7	$6.7 \times 10^4$	1.00
9	6	84	6	90	6	542.1	$5.6 \times 10^6$	1.83
	7	36	21	24	100	506.4	$2.4 \times 10^6$	1.02
	8	9	56	5	99	368.4	$6.0 \times 10^5$	1.00
	9	1	126	5	22	112.4	$6.7 \times 10^4$	1.00
10	6	210	6	150	7	1215.0	$1.4 \times 10^7$	1.58
	7	120	21	21	100	1673.8	$8.0 \times 10^6$	1.05
	8	45	56	6	100	1803.9	$3.0 \times 10^6$	1.02
	9	10	126	2	100	997.7	$6.7 \times 10^5$	1.01
	10	1	252	2	70	230.8	$6.7 \times 10^4$	1.00

ID, 点 ID, 複比の列  $r_{(0)}, \dots, r_{(m, C_5-1)}$  の組が一つの要素)の数の合計であり、実際にはこれに要素一つ当りの大きさを掛けた記憶容量 (実験では  $(24 + m C_5)$  バイト)が必要となる。 $l$  はハッシュ表においてリスト長が 0 でないもののリスト長の平均である。表 1~表 4 には各  $n, m$  の組について、最も高い精度が得られた  $k$  を用いた結果のみを示している。また、参考のため、各  $n, m$  における組合せの数  ${}_n C_m$  及び  ${}_m C_5$  も示してある。

まず、精度について考察する。全体として、撮影角度が小さくなるにつれて精度が低下した。これは、射影ひずみが大いいと近傍点の変化が大きくなり、特徴

表 4 パラメータと性能の関係 (撮影角度 30°)  
Table 4 The relation between parameters and performance (angle 30°).

$n$	$m$	${}_n C_m$	${}_m C_5$	$k$	精度 [%]	処理時間 [msec.]	リスト数	$l$
6	6	1	6	40	5	20.6	$6.7 \times 10^4$	1.18
7	6	7	6	40	3	60.3	$4.7 \times 10^5$	1.82
	7	1	21	8	7	31.3	$6.7 \times 10^4$	1.03
8	6	28	6	180	4	147.7	$1.9 \times 10^6$	1.08
	7	8	21	15	12	138.1	$5.3 \times 10^5$	1.01
	8	1	56	9	6	60.1	$6.7 \times 10^4$	1.00
9	6	84	6	90	5	611.0	$5.6 \times 10^6$	1.83
	7	36	21	22	25	566.6	$2.4 \times 10^6$	1.02
	8	9	56	5	20	413.3	$6.0 \times 10^5$	1.01
10	9	1	126	5	1	125.6	$6.7 \times 10^4$	1.00
	6	210	6	210	4	1183.4	$1.4 \times 10^7$	1.31
	7	120	21	24	39	1885.0	$8.0 \times 10^6$	1.04
	8	45	56	6	45	2019.4	$3.0 \times 10^6$	1.02
10	9	10	126	3	18	1115.3	$6.7 \times 10^5$	1.00
	10	1	252	2	4	258.1	$6.7 \times 10^4$	1.00

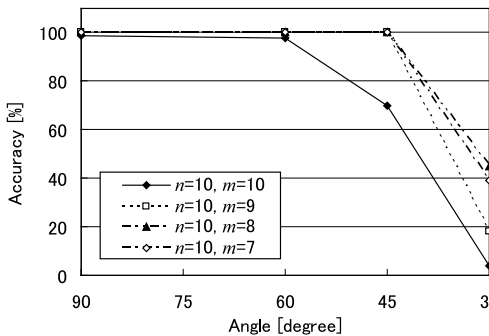


図 12 撮影角度と検索精度の関係

Fig. 12 Relation between angles and accuracy.

量の安定性を保つことができなくなるためと考えられる。ただし、図 12 に示すように  $n$  と  $m$  の差が大きいと精度の低下が抑えられた。これは、 $n - m$  が許容される特徴点の消失数であることによる。しかし、 $m = 6$  のときは  $n$  の値が大きくなるにつれて逆に精度が低下している。これは、特徴量に含まれる複比の数  ${}_m C_5$  が小さいので識別性が低く、特徴点から計算される特徴量の数  ${}_n C_m$  が大きくなると異なる特徴点からも同じ特徴量が得られるようになることが原因であると考えられる。

精度の問題に関しては、ノイズや射影ひずみによる近傍点の変化やハッシュのインデックスを求める際の特徴量の離散化誤差などが影響していると考えられる。この点を明確にするため、ハッシュを用いない特徴量の最近傍探索などの手法と比較する必要がある。ただし、最近傍探索は計算量が膨大<sup>(注1)</sup>であるため、実用

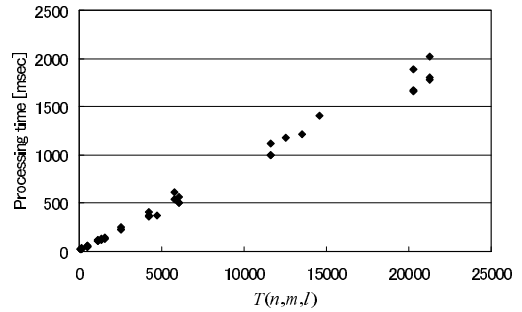


図 13  $T(n, m, l)$  と処理時間の関係

Fig. 13 Relation between  $T(n, m, l)$  and processing time.

上の比較対象とはならない。

次に、処理時間について考察する。図 8 に示した検索アルゴリズムの構造から、処理時間は特徴量計算及びリスト処理に要する時間と、それらの繰返し回数によって定まると考えられる。特徴量計算に要する時間は、おおむね特徴量を構成する複比の数である  ${}_m C_5$  に比例し、リスト処理に要する時間は平均リスト長  $l$  に比例する。また、それらの繰返し回数は  ${}_n C_m \cdot m$  である。以上のことから、パラメータ  $n, m$  と平均リスト長  $l$  から求められる処理時間の目安として  $T(n, m, l)$  を以下の式で定義する。

$$T(n, m, l) = {}_n C_m \cdot m \cdot ({}_m C_5 + \alpha l) \quad (4)$$

ここで、 $\alpha$  は特徴量計算に対するリスト処理の時間の重みである。図 13 に  $\alpha = 3$  としたときの  $T(n, m, l)$  と処理時間のグラフを示す。図 13 より、 $T(n, m, l)$  と処理時間はほぼ比例することが分かる。したがって、処理時間は式 (4) のような形で  $n, m, l$  の影響を受けると考えられる。

最後に、記憶容量について考察する。表 1~表 4 では、ハッシュ表に記憶されるリスト数は  ${}_n C_m$  に比例したものになっている。これは、各点の特徴量ごとに登録が行われるためである。上述したように、高い安定性を実現するには  $n - m$  の値を大きくする必要があるが、これは同時に、必要となる記憶容量の増大をもたらすといえる。

### 3.3 実験 2: 量子化レベル $k$ と性能の関係

提案手法の性能は、量子化レベル  $k$  の値によっても変化する。そこで、データベース B 及び検索質問 2 を

(注1): 特徴点数を実験での平均値 630,  $n = 8, m = 7$  とすると、データベースのサイズが 100 ページであっても、単純な逐次比較では特徴量の比較回数は提案手法の 50 万倍にもなる。



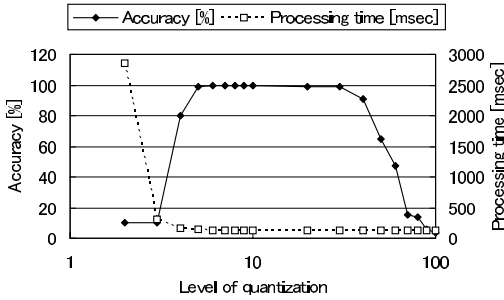


図 14 量子化レベルと検索精度・処理時間の関係

Fig. 14 Relation between level of quantization and accuracy, and that between level of quantization and processing time.

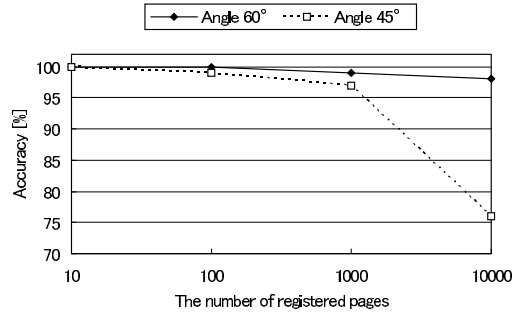


図 15 登録ページ数と検索精度の関係

Fig. 15 Relation between the number of registered pages and accuracy.

用いてパラメータを  $n = 8$ ,  $m = 7$  と設定し,  $k$  と精度及び処理時間の関係を調べた. 結果を図 14 に示す.

まず精度について考察する.  $k$  が小さいときの精度は低く,  $k$  の増加に伴って精度も急激に上昇した. これは, 2.4.5 で述べたように  $k$  が小さいときは特徴量の識別性が低いため, 正しく対応する文書とそうでないものを区別できないことが原因と考えられる. また,  $k$  が大きくなりすぎると, 精度が急激に低下した. これは, 特徴量の安定性が低下するためと考えられる.

次に処理時間について考察する. 処理時間は,  $k$  の増加に伴って急激に減少した後, ほぼ一定の値に保たれている. これは,  $k$  が小さいときは特徴量の識別性が低くハッシュ表での衝突が多くなるため, 検索時のハッシュアクセスに時間がかかることが原因と考えられる.

以上のことから, 高速かつ高精度な検索を実現するには  $k$  の値を適切に設定することが必要といえる.

### 3.4 実験 3: 登録ページ数と検索精度の関係

データベース A~D を用いて登録ページ数を 10 から 10,000 まで変化させたときの登録ページ数と検索精度の関係を調べた. 検索質問は 2 及び 3 を用いた. また, パラメータは両方の検索質問で統一し,  $n = 8$ ,  $m = 7$  とした. このとき,  $k$  の値は表 2, 表 3 に示す値を用いた.

実験結果を図 15 に示す. ページ数の増加に伴って, 精度が低下していくことが分かる. これは, データベースが大きいと, 同じ特徴量をもつ異なる文書が登録されている可能性が高くなるためと考えられる.

また, 検索質問 2 に比べて検索質問 3 では精度が低くなっている. これは, 射影ひずみが大きいため近傍点の変動が大きく, 特徴量を安定に得ることが困難であることが原因と考えられる.



図 16 検索失敗例

Fig. 16 An example of erroneous cases.

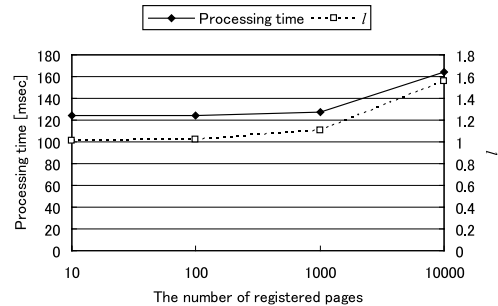


図 17 登録ページ数と検索速度・リスト長の関係

Fig. 17 Relation between the number of registered pages and processing time, and that between the number of registered pages and  $l$ .

検索に失敗した検索質問の例を図 16 に示す. この画像のように, ページの大部分を図表が占め, テキストの量が少ない検索質問で検索に失敗した. これは, 得られる特徴点の数が少ないために検索において正解の文書が十分な得票数を得られないことが原因と考えられる. この点については, 特徴点抽出について新たな

な処理を追加するなどの改良が必要である。

### 3.5 実験4：登録ページ数と処理時間の関係

登録ページ数が処理時間にどのように影響するかを調べた。データベースはA~D、検索質間は2、パラメータは $n = 8, m = 7, k = 12$ を用いた。結果を図17に示す。登録文書数が増加するに従って処理時間は少しずつ増加することが分かる。3.2で述べたように、処理時間はパラメータ $n, m$ と平均リスト長 $l$ に影響される。この実験ではパラメータ $n, m$ は固定されているので、平均リスト長 $l$ を同図に示す。登録ページ数の増加に伴って平均リスト長が増加していることから、ハッシュ表で衝突が増加していることが分かる。これが処理時間の増加の理由と考えられる。

## 4. むすび

本論文では、特徴点の局所的配置から求めた特徴量と、ハッシュ表を用いた投票によって検索を行う文書画像検索法を提案した。提案手法の特徴は、特徴量の計算を各特徴点の局所的配置に基づいて求めることによる低い計算量(特徴点数 $N$ に対して $O(N)$ )と、安定性と識別性を両立した特徴量による高い検索精度である。実験により、提案手法によって大規模なデータベースからの高速かつ高精度な文書画像検索が可能になることが示された。

今後の課題としては、最近傍探索との比較を通じた精度の改善、日本語文書への対応、文書に限らないシーン中のオブジェクトの検索手法の考案が挙げられる。また、他の不変量の利用についても検討が必要と考えられる。

謝辞 本研究の一部はJST 実用化のための可能性試験、大川情報通信基金研究助成、大阪府立大学大学院工学研究科FI 推進研究奨励研究費の助成による。

## 文 献

- [1] D. Doermann, "The indexing and retrieval of document images: A survey," *Computer Vision and Image Understanding: CVIU*, vol.70, no.3, pp.287-298, June 1998.
  - [2] J.J. Hull, "Document image matching and retrieval with multiple distortion-invariant descriptors," in *Document Analysis Systems*, ed. A.L. Spitz and A. Dengel, pp.379-396, World Scientific, Singapore, 1995.
  - [3] D. Doermann, H. Li, and O. Kia, "The detection of duplicates in document image databases," *Proc. 4th International Conference on Document Analysis and Recognition*, pp.314-318, Ulm, Germany, 1997.
  - [4] D. Doermann, J. Liang, and H. Li, "Progress in camera-based document image analysis," *Proc. 7th International Conference on Document Analysis and Recognition*, pp.606-616, Edinburgh, Scotland, 2003.
  - [5] P. Clark and M. Mirmehdi, "Recognising text in real scenes," *International Journal on Document Analysis and Recognition*, vol.4, no.4, pp.243-257, Aug. 2002.
  - [6] 黄瀬浩一, 大町真一郎, 内田誠一, 岩村雅一, "カメラを用いた文字認識・文書画像解析の現状と課題," *信学技報*, PRMU2004-246, March 2005.
  - [7] S. Pollard and M. Pitu, "Building cameras for capturing documents," *International Journal on Document Analysis and Recognition*, vol.7, no.2-3, pp.123-137, July 2005.
  - [8] H.J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE Comput. Sci. Eng.*, vol.4, no.4, pp.10-21, Oct. 1997.
  - [9] <http://sourceforge.net/projects/opencvlibrary/>
  - [10] T. Suk and J. Flusser, "Point-based projective invariants," *Pattern Recognit.*, vol.33, no.2, pp.251-261, Feb. 2000.
- (平成17年12月14日受付, 18年3月14日再受付)



中居 友弘 (学生員)

平16 阪府大・工・情報卒。平18 同大学院修士課程了。現在、同大学院博士課程に在籍。文書画像処理に関する研究に従事。



黄瀬 浩一 (正員)

昭61 阪大・工・通信卒。昭63 同大学院博士前期課程了。同年同大学院博士後期課程入学。平2 阪府大・工・電気助手。現在、同大学院工学研究科教授。博士(工学)。平12年~13 ドイツ人工知能研究センター客員教授。文書画像解析, 情報検索などの研究に従事。情報処理学会, 人工知能学会, 電気学会, IEEE, ACM 等各会員。



岩村 雅一 (正員)

平10 東北大・工・通信卒。平15 同大学院博士課程了。同年同大学院工学研究科助手。平16 阪府大大学院工学研究科助手。現在に至る。博士(工学)。パターン認識, コンピュータビジョンに関する研究に従事。IEEE, 情報処理学会, Pattern Recognition Society 各会員。