

# 局所特徴量のハッシングに基づく大規模画像検索

黄瀬 浩一<sup>†</sup> 岩村 雅一<sup>†</sup> 中居 友弘<sup>†,††</sup> 野口 和人<sup>†</sup>

<sup>†</sup> 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

<sup>††</sup> 日本学術振興会特別研究員

E-mail: <sup>†</sup>{kise,masa}@cs.osakafu-u.ac.jp, <sup>††</sup>{nakai,noguchi}@m.cs.osakafu-u.ac.jp

あらまし 検索質問を画像とする画像検索は、バーコードの代替や著作権保護など、数多くのアプリケーションを持つ重要なタスクである。一般にカメラ付き携帯電話などのデバイスで撮影される検索質問は、データベースの画像と同じものを写していても、撮影条件の変動によりデータとしては全く異なるものとなるため、検索は容易ではない。この問題に対処するため、近年、局所特徴量を用いた検索が注目されている。局所特徴量を用いた検索は、撮影条件にあまり影響を受けない検索を可能とする反面、索引付けに用いられる特徴量（特徴ベクトル）の数が膨大となるため、処理時間に関する問題が生じる。本稿では、この問題を解決する方策として、我々が取り組んでいるハッシュ表を用いた検索手法について述べる。検索対象としては、風景や人物を撮影した一般の画像に加え、文書画像という全く性質の異なるものを取り上げ、各々に有効な局所特徴量を示すとともに、両者の検索に共通する考え方を述べる。

キーワード 画像検索，物体認識，文書画像検索，局所特徴量，不変量，ハッシュ表

## Large-Scale Image Retrieval Based on Hashing of Local Features

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

<sup>†</sup> Graduate School of Engineering, Osaka Prefecture University

1-1, Gakuencho, Naka, Sakai, Osaka 599-8531 Japan

<sup>††</sup> Japan Society for the Promotion of Science (JSPS)

E-mail: <sup>†</sup>{kise,masa}@cs.osakafu-u.ac.jp, <sup>††</sup>{nakai,noguchi}@m.cs.osakafu-u.ac.jp

**Abstract** Image retrieval with query images is a task to find their corresponding images (near duplicates) from the database. It is an important task since it has many applications such as a replacement of barcodes and copyright protection of images. The task of retrieval is not trivial since query images are often far different from their corresponding images in the database. This is due to various imaging conditions, e.g., occlusion, perspective distortion, different lighting conditions, defocus and motion blur. In order to solve this problem, retrieval based on local features attracts considerable attention. In this method, images are indexed using feature vectors extracted from local regions of images. Although they allow us the retrieval relatively robust to imaging conditions, they pose a problem of processing time since the number of local features is huge. In this report, we describe our new retrieval methods based on hashing for solving the above problem. As images in the database, we deal with ordinary images about scene, objects and people, etc., as well as document images that are completely dissimilar to ordinary images. Local features for each type of images and the common notion for successful retrieval are shown.

**Key words** Image retrieval, Object recognition, Document image retrieval, Local feature, Invariant, Hash table

### 1. ま え が き

大量の画像データが我々の周囲に溢れている。インターネット上の画像共有サイト内は言うに及ばず、個人のPC内でもすでに手作業での管理が追いつかないほど膨大な量の画像を持つ人も多い。画像の内容としては、物、風景、人物などの一般の画像だけではなく、省スペース化のために画像化された文書

(文書画像)など、多様な性質のものが混在している。これらの画像を有効利用するために必要な基本的処理の一つは検索であろう。実際、画像共有サイトでは、キーワードを検索質問とする検索(キーワード検索)がサポートされている。また文書画像については文字認識を施せば電子文書と同様にキーワード検索が可能となる。

それでは、キーワード検索があれば画像は十分活用できるで

あろうか。最近のデジタルカメラやカメラ付き携帯電話の普及に伴って、全く異なるタイプの検索も必要であるとの認識が広がりつつある。これは、画像を検索質問として、その画像に写っているものと同じものが写った画像を検索するという処理である。このとき、データベースから画像を検索することは中間的な目標でしかなく、画像を介して別種の情報にアクセスすることが目的となる。具体的には、例えばカタログに掲載されている商品の写真を撮影して検索質問とし、同じ商品が写った画像を検索することによって、その商品についてより詳しい情報を得ることが考えられる。また、観光地のランドマークを撮影し、観光情報を得ることも可能である。文書の場合であれば、その文書に関連付けられた別の情報を得ることになる。このような処理は、画像の検索と捉えられるほか、画像中に写った特定の物体の認識とも捉えることができる。また、上記の例にあげた処理は、バーコードの代替とも捉えることができる。

このような画像を検索質問として「同じ物体が写った画像」を検索する処理の実現には、いくつかのハードルを乗り越えなければならない。

最も大きな問題は、同じ物が写っているという共通性はあるものの、データベース中の画像と検索質問の画像は大きく異なることが多いという点である。この問題は、撮影条件の違い、即ち撮影範囲や角度、照明条件、解像度などの違いから生じる。従って画像検索の手法は、これらの表面的な相違を克服する必要がある。幸い、これらの相違にあまり影響を受けない安定した画像特徴量として、局所特徴量と呼ばれるものが提案されている。局所特徴量は、画像の一部分から抽出される特徴量であり、撮影条件が異なってもほぼ同一の部分から類似の特徴量が抽出できるという好ましい性質がある。

ところが、局所特徴量には次のような問題点もある。まず画像から抽出される特徴量の数が1画像あたり数百から数万と、非常に膨大になる点である。例えば1画像あたり千個の特徴量が取り出されるとき、10万画像のデータベースは一億の特徴ベクトルから構成されることになる。また、特徴量同士の全数照合による検索を考えると、照合回数は一十億回となる。特徴量は通常、数十から数百次元の特徴ベクトルとして表されるため膨大な計算が必要となり、実際の計算時間では照合を終えることができない。

本稿では、このような問題点を解決するために、我々がこれまで開発してきた2つの手法について俯瞰的に紹介する。一つは、一般の画像を対象とした検索法[1]であり、局所特徴量としてPCA-SIFT[2]と呼ばれるものを用いるものである。データベース中の画像数を10万画像まで増加させても、検索質問あたり100ミリ秒で照合が可能である。もう一つは文書画像を対象とした手法である[3]。局所特徴量としては独自に開発したものをを用いている。これは、幾何学的不変量を組み合わせで得られるものであり、撮影角度が変わっても安定して同じ特徴量が得られる。1万ページのデータについて100ミリ秒程度で検索が可能となっている。両者に共通する高速化の仕組みは、ハッシュ表を用いることである。これにより、データベース中のサイズにあまり依存することなく、高速検索が可能となる。

## 2. 局所特徴量とその照合

### 2.1 大域 vs. 局所

画像の照合に用いられる特徴量は大きく大域特徴量(global feature)と局所特徴量(local feature)に分類できる。大域特徴量とは、画像全体から抽出される特徴量であり、Content-Based Image Retrieval (CBIR) に用いられるカラーヒストグラムがその一例である。一方、局所特徴量とは、画像の一部分から抽出される特徴量である。最も著名なものはSIFT(Scale-Invariant Feature Transform) [4]であろう。局所特徴量を抽出するには、どの領域から特徴量を取り出すのかを決定し、その領域から特徴量を抽出するという2段階の処理が必要になる。前者を担うプログラムをdetector、後者を担うプログラムをdescriptorと呼ぶこともある。大域、局所の両者とも特徴量はベクトル(特徴ベクトル)として表現される。これを照合することによって、画像を検索することが可能となる。

局所特徴量のdetectorに対応する処理が大域特徴量の抽出では不要であるため、大域特徴量の抽出は計算量的に有利である。また、特徴量が画像1枚あたり1つであるため、画像の索引付けに用いるデータ量としても大幅に少なくて済むという利点もある。一方で、画像の一部分が隠れ(occlusion)によって得られない場合、大域特徴量ではもはや同じ値を得ることは不可能となるという問題もある。局所特徴量を用いると、隠れによって得られない特徴量があっても、隠れていない部分から依然として同じ特徴量を得ることができるという利点がある。本研究では、このような局所特徴量の利点が、画像を検索質問とする画像検索に必須であると考え、局所特徴量を用いる。

一般に用いるべき局所特徴量は、対象とする画像に大きく依存する。ここでは、風景や物体の写真、ポスターなどの画像を一般画像と呼ぶことにする。以下では、まず、局所特徴量の照合について述べたあと、一般画像と文書画像の各々について特徴量の詳細を述べる。

### 2.2 局所特徴量の照合と画像検索

データベースには、検索対象の画像から得た局所特徴量が記録されているとする。一方、検索質問からも複数の局所特徴量が取り出される。このような問題設定において、画像を検索する最も単純な方法は以下の通りである。検索質問から得た局所特徴量の各々について、データベースから最も類似するものを探し出す。データベースの局所特徴量は、どの画像から得たものであるのかが分かるため、検索質問の局所特徴量について、画像との対応が得られる。これを投票と考えて、最大得票数の画像を検索結果とすればよい。

以下に述べるように、局所特徴量は特徴ベクトルとして表されるため、最も類似した特徴ベクトルとは、例えば特徴ベクトル同士のユークリッド距離が一番小さいものとして定義することができる。この場合、局所特徴量の照合は、特徴ベクトルの最近傍探索となる。

### 2.3 一般画像

一般画像に対する局所特徴量として、本研究ではPCA-SIFTを用いる。SIFTが128次元の特徴ベクトルであるのに対して、



(a)



(b)

図 1 局所領域の不変性

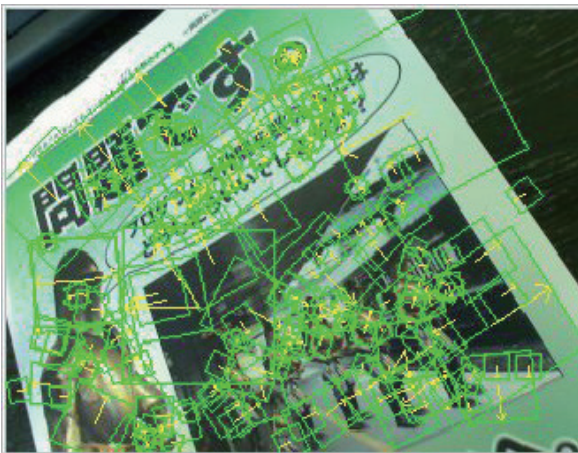


図 2 実際の局所領域の例

PCA-SIFT は主成分分析により低次元 (具体的には 36 次元) へと次元を圧縮した特徴ベクトルとなっている。

SIFT と PCA-SIFT は同じ detector に基づくものである。図 1 に示すように、画像の相似変換 (並進, 回転, 拡大縮小) に対して不変な領域を取り出せるという特徴を持つ。カメラで対象物を撮影する場合には、射影変換と呼ばれる、相似変換よりも自由度の高い変換が加わるが、角度が急でないならば (正面から撮影する場合を  $90^\circ$  とするとき,  $60^\circ$  程度までであれば), 同様に不変な領域が得られることも多い。図 2 に PCA-SIFT で得られる実際の局所領域を示す。ここで、四角形とその内部の矢印は、局所領域とその向きを表している。このように実際には、様々な大きさの重なり合った局所領域が多数得られる。

どのような特徴ベクトルがいくつ抽出されるのかは対象とする画像に大きく依存する。以下では典型的な画像 (スナップ写真のようなもの) を対象とした場合について述べる。

抽出される特徴ベクトルの数は、おおよそ以下の通りである。VGA サイズの原画像 (データベース格納用) からならば 2,000 個程度, VGA サイズの画像をプリントアウトして、それを VGA あるいは QVGA のサイズで撮影した画像 (検索質問用) からならば 400 ~ 100 個程度である。従って、例えば 10 万

画像のデータベースを考える場合、特徴ベクトルの数は 2 億となる。

いま、検索質問から得られる特徴ベクトルの数を 200 個とし、各々について最近傍探索による照合を行うことを考えると、必要な照合回数は 400 億回という膨大な回数となる。これは、最近のコンピュータを用いて照合しても、検索質問あたり 9 時間以上の時間を要するという量である。従って、高速化が必須であることが分かる。

もう一つの重要な問題は、特徴ベクトルの識別能力に関するものである。検索質問の特徴ベクトルに対して、データベース中の 2 億もの特徴ベクトルの中から最近傍となるものを選んだとき、それが同じ画像から得られた特徴ベクトルになるのだろうか。写真などの平面物体の認識というタスクに対して、実際に処理を行ってみると PCA-SIFT は十分な識別能力があることが分かる。すなわち最近傍探索を行うことができれば、検索に十分な数の正しい最近傍 (正解画像への投票) が得られる。

なお、最近の成果によると、特徴ベクトルは次元あたり 32 ビットの実数表現でなくてもよく、次元あたり 2 ビットの表現にスカラー量子化しても、認識結果には殆ど影響を及ぼさないことがわかっている [5]。逆にいえば、それほど識別能力が高いことになる。

#### 2.4 文書画像

上記の PCA-SIFT を含めて、一般画像に対して有効な局所特徴量は、必ずしも文書画像に対して有効であるとは限らない。この理由は明確であり、VGA あるいは QVGA という解像度で文書を撮影すると、どの文書の局所領域も同じような模様を持つものにしか見えないことである。

この問題に対処するためには、画像の解像度を高めるか、あるいは新しい特徴量を考案するかのいずれかが必要となる。前者については、画像の解像度を 200dpi 程度 (vga の約 4 倍) まで高めると良いことが分かっている。しかし、この場合、特徴抽出に必要な処理時間が膨大になるという問題点も生じる。そこでここでは、文書画像に適した新しい局所特徴量について述べる。

一般画像に対する局所特徴量は、局所領域から識別に十分な特徴量を取り出すものである。大半の局所領域は、画像のサイズに対して小さい領域となる傾向にある。一般画像のように局所領域内の画像が十分不均質であれば問題ないが、文書画像のように均質な場合は識別性に欠ける。この問題を解決する一方策は、特徴量を取り出す範囲をより広くすることである。

本研究では、図 3(a) に示すような単語領域の重心や、図 3(b) に示すような黒画素連結成分の重心を考え、その局所的な配置を特徴とする。カメラで対象を撮影することを考えると、配置は撮影角度の変化によって生じる幾何学的変動に不変な方法で表現できることが望ましい。そこで本手法では、幾何学的不変量を用いて配置を表現する。具体的には、以下の通りである。

本手法では、特徴点の近傍  $m$  点の配置をベクトルに変換して特徴量とする。まず、不変量としては、4 点の配置で定まるアフィン不変量を用いる。いま、点  $A, B, C, D$  の 4 点があるとき、アフィン不変量として、3 角形の面積比

resultant words  
 ters of their bo  
 o dimensional d  
 introduced to tl

(a) 単語領域の重心

活用形は語の接続条  
 トや自立語の活用語尾  
 用形は文節内の自立  
 する。すなわち活用

(b) 連結成分の重心

図 3 文書画像の特徴点

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

を用いる。ここで、 $P(X, Y, Z)$  は点  $X, Y, Z$  を頂点とする 3 角形の面積である。特徴量はこれらを複数用いて定義されるベクトルである。具体的には、 $m$  個の点から 4 個を選ぶ  ${}_m C_4$  個の組み合わせをすべて考え、それらを要素とするベクトルを特徴量とする。

検索のためにはデータベースの文書画像と検索質問の文書画像から同じ特徴量が得られなければならない。ところが、撮影条件によっては、同じ  $m$  個の特徴点が得られないことがある。例えば、近傍  $m$  点は撮影角度によって変化するし、連結成分の大きさも変化する。そこで本手法では、近傍  $n$  点 ( $n > m$ ) を考え、 $n$  点の中から  $m$  点を選ぶ組み合わせをすべて試す。これによって、 $(n - m)$  点が正しく得られない場合でも、同じ特徴量を得ることができる。

実際の特徴量の数は、一般画像の場合と同様に対象文書に依存する。2 段組の英文文書の場合であれば、おおよそ以下の通りである。データベースの文書画像からは 600 個程度の特徴点が見られる。パラメータの値を  $n = 8, m = 7$  とすると、特徴ベクトルの次元数は  ${}_7 C_4 = 35$  次元、1 点あたり  ${}_8 C_7 = 8$  個の特徴ベクトルが見られる。1 万画像を対象とする場合、データベース中の特徴ベクトルの数は、約 5 千万 ( $600 \times 8 \times 10,000$ ) となる。検索質問からは約 400 程度の特徴点が見られ、同様の計算によって特徴ベクトルの数は 3200 ( $400 \times 8$ ) となる。従って、単純な最近傍探索の場合の比較回数は、一般画像の場合と同様に、約 1500 億回という膨大な数となる。

### 3. ハッシングによる高速化

#### 3.1 近似最近傍探索

上記のような膨大な数の特徴ベクトルを対象として実際的な時間で最近傍探索を行うためには、データ構造に工夫が必要である。最も基本的なデータ構造としては  $k$ - $d$  木がある。しかしながら、特徴ベクトルが高次元になるにつれて、特徴ベクトルの分布には球面集中現象が見られるため、 $k$ - $d$  木の有効性が損なわれることが知られている。

この問題を解決する一つの方策は、厳密な最近傍を求めることを諦め、近似を行うことである。幸い、前述の最も単純な画像検索を考えると、結果は数多くの票を用いた投票によって決まるため、誤った画像の得票数が正解の画像の得票数を上回らない限り、正解の画像を検索することができる。

近似最近傍探索は、距離計算の対象となる特徴ベクトルを大胆に絞り込むことによって高速化を図るものである。絞り込みの際に、最近傍の特徴ベクトルも誤って除外してしまうことのある程度許容できるタスクでは、大幅な高速化が可能である。

近似最近傍探索の代表的な方法は、ANN (Approximate Nearest Neighbor) [6] と LSH (Locality Sensitive Hashing) [7] の 2 つであろう。ANN は前述の  $k$ - $d$  木を基礎とする木構造ベースの手法である。一方、LSH はハッシュ表を用いた手法である。両者とも優れた手法であるが、基本的には特徴ベクトルのデータが主記憶上にあることが前提となっているため、扱うことのできる特徴ベクトルの数をそれほど増やすことはできない。例えば、主記憶 16GB 程度の計算機では、10 万枚の一般画像を扱うことができない。

LSH が大量にメモリを必要とする原因は、複数のハッシュ表を用いることにある。LSH は、複数のハッシュ関数に対応した複数のハッシュ表を用いることによって、最近傍の検索漏れを低く抑えている。したがって、検索漏れを抑える別の仕組みがあれば、ハッシュ表を一つに絞ることが可能となり、メモリの問題は解決する。

ハッシュ表を一つに絞る仕組みは、一般画像と文書画像で異なる。以下、順に述べる。

#### 3.2 一般画像の場合

PCA-SIFT により得られる 36 次元特徴ベクトルを  $x = (x_1, \dots, x_{36})$  とする。ここで、1 次元目から順に、より大きな固有値に対応するものとする。本手法では、1 次元目から  $d$  次元目までを用いてハッシュ表に格納する。いま、

$$u_i = \begin{cases} 1 & \text{if } x_i - \mu_i \geq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

で定められる  $d$  次元のビットベクトルを考える。ここで  $\mu_i$  は、データベースに収められた  $i$  次元目の値の平均値である。このビットベクトルを用いて、 $x$  のハッシュ関数  $h(x)$  を、

$$h(x) = \left( \sum_{i=1}^d u_i 2^{(i-1)} \right) \bmod H_{\text{size}} \quad (2)$$

のように定める。ここで、 $H_{\text{size}}$  はハッシュ表におけるピンの

個数である。

このハッシュ関数を用いて特徴ベクトルを格納していくと、同じハッシュ値を持つ特徴ベクトルが出てくる。基本的にはチェーン法によって格納することとする。ただし、その数が多くなりすぎると、処理時間やメモリ量、検索精度に悪影響を及ぼす。そこで上限  $c$  を設け、登録数が  $c$  を超える場合には、全体をハッシュ表から削除するとともに、以後同じピンには特徴ベクトルを登録しないことにする。

近似最近傍探索の方法は以下の通りである。上記のハッシュ関数を検索質問ベクトル  $q = (q_1, \dots, q_{36})$  にも適用してハッシュ表にアクセスする場合、特徴ベクトルのある次元  $i$  の値  $q_i$  が閾値  $\mu_i$  に近ければ、最近傍は異なるビットベクトルによってハッシュ表に格納されている可能性がある。この問題を回避する単純な方法は、そのような「疑わしい」場合に、当該次元を反転 (0 ならば 1, 1 ならば 0 とする) した別のビットベクトルも用いてハッシュ表にアクセスすることである。具体的には、 $e$  を閾値として  $|q_i - \mu_i| \leq e$  を満たすならば反転も試す。これは、特徴ベクトルの値に応じて異なるハッシュ関数を用いてハッシュ表にアクセスすることを意味する。ただし、このような処理を際限なく適用すると、適用するハッシュ関数の数が 2 の冪乗で増えて処理時間に問題が生じる。そこで、反転を適用する次元数を、特徴ベクトル一つに対して  $b$  個に制限する。具体的には、 $i = 1$  から反転の条件を満たすかどうかをチェックし、 $b$  個の上限に達した場合には以後の処理を行わないこととする。これにより、近似最近傍探索の時間を低く抑えたままで、柔軟な検索が可能となる。

### 3.3 文書画像の場合

2.4 で述べたように、文書画像の場合には、特徴点の消失に対応するため、複数の特徴ベクトルを生成する。このような仕組みを用いることで柔軟性が得られるため、一般画像のように複数のハッシュ関数を用いることは行わない。

一方、文書画像の局所特徴量の特性から、一般画像の場合のようにビットベクトルを用いたハッシュ関数では識別能力に不足が生じてしまう。そこで各次元を 2 レベルではなく  $k$  レベルに量子化してハッシュ表に格納する。ハッシュ関数  $h(x)$  は、特徴ベクトル  $x$  を  $x = (x_{(1)}, \dots, x_{(i)}, \dots, x_{(mC_f)})$  とするとき、以下の通りである。

$$h(x) = \left( \sum_{i=1}^{mC_f} x_{(i)} k^{(i-1)} \right) \bmod H_{size} \quad (3)$$

ここで、 $H_{size}$  はハッシュ表におけるピンの個数である。このハッシュ関数は、特徴ベクトルの索引付けに用いられる。

## 4. 実験

一般画像と文書画像について行った実験の結果を順に述べる。

### 4.1 一般画像検索

flicker からダウンロードした 10 万枚の画像をデータベースに格納するとともに、そのうち 500 枚をランダムに選択して印刷し、撮影角度や範囲を 4 通り変更して撮影することにより 2,000 画像の検索質問を作成した。画像のサイズはデータベー

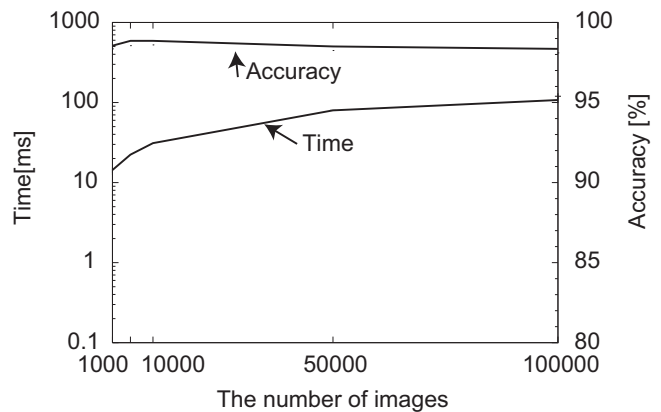


図 4 一般画像の検索結果

表 1 文書画像検索の結果

DB のサイズ (画像数)	精度 [%]	処理時間 [ms]
100	99	109.3
1000	99	109.5
10000	98	119.6

ス、検索質問ともほぼ VGA とした。各種パラメータの値は  $c = 10, d = 28, b = 10, e = 400$  である。また、実験に用いた計算機は、CPU が AMD Opteron 2.8GHz、メモリが 32GB のものである。

実験結果を図 4 に示す。ここで、横軸はデータベースのサイズ (格納された画像数)、縦軸は精度と処理時間である。精度 (accuracy) とは検索質問に対応する画像が正しく検索された割合、処理時間とは検索質問あたりに要した時間 (ただし、PCA-SIFT による特徴抽出の時間は含まない) を表す。この図からわかるように、データベースのサイズが増加しても一定の精度が得られている。処理時間については、線形よりも緩やかに増えている。10 万画像のデータベースの結果は、精度 98.4%、処理時間 107ms であった。

### 4.2 文書画像検索

英文の国際会議プロシーディングスから得た PDF ファイル 1 万ページを対象としてデータベースを作成した。また、検索質問としては、正面を  $90^\circ$  としたとき、 $60^\circ$  から撮影した 100 枚を用いた。画像の例を図 5 に示す。画像の解像度はデータベース側が 200dpi 程度 (画像サイズは  $1,700 \times 2,200$ )、検索質問が 630 万画素 ( $2,048 \times 3,072$ ) である。処理に用いたパラメータは  $n = 8, m = 7$  である。また、実験に用いた計算機は、CPU が AMD Opteron 2.8GHz、メモリが 16GB のものである。

結果を表 1 に示す。同様に処理時間には特徴抽出の時間は含まれない。この結果から、精度と処理時間はともに、ほぼデータベースのサイズによらず一定であることがわかる。

## 5. むすび

本稿では、局所特徴量のハッシングという手法を用いた大規模画像検索について述べた。局所特徴量をハッシュに収める方法と、それを検索する方法を工夫することによって、多数の画像から得られる膨大な局所特徴量との照合も高速に行え、デー

本研究の一部は科学研究費補助金(基盤研究(B)19300062)(特別研究員奨励費19・7621)の補助による。

文 献

- [1] 野口和人, 黄瀬浩一, 岩村雅一, “局所記述子に基づく物体認識のためのメモリ削減の実験的検討,” 画像の認識・理解シンポジウム(MIRU2008) 論文集, pp.251-258, July, 2008.
- [2] Y. Ke and R. Sukthankar, Pca-sift: A more distinctive representation for local image descriptors, CVPR2004, Vol. 2, pp.506-513, 2004.
- [3] 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リアルタイム文書画像検索とその拡張現実への応用,” 電子情報通信学会技術研究報告, vol.106, no.PRMU-229, pp.41-48, Sept., 2006.
- [4] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol.60, no.2, pp.91-110, 2004.
- [5] 野口和人, 黄瀬浩一, 岩村雅一, “局所記述子に基づく物体認識のためのメモリ削減の実験的検討,” MIRU2008, 第1巻, pp.321-324, July, 2008.
- [6] S. Arya, D. M. Mount, R. Silverman and A. Y. Wu, “An optimal algorithm for approximate nearest neighbor searching,” Journal of the ACM, vol.45, no.6, pp.891-923, 1998.
- [7] A. Andoni, M. Datar, N. Immorlica, P. Indyk and V. Mirrokni, Locality-sensitive hashing using stable distributions, Nearest-Neighbor Methods in Learning and Vision (Eds. by G. Shakhnarovich, T. Darrel and P. Indyk), The MIT Press, pp.61-72, 2005.

Let line figure 5(a) by imposing the following constraint on  $P$  and  $P'$ .

- The origins of  $Z$  and  $Z'$  such that  $(0,0)$  maps to  $(0,0)$ .
- Project the epipoles to infinity such that  $a$  and  $a'$  map to  $(0,0)$ .

These conditions are satisfied if  $P$  and  $P'$  are of the form

$$P = \begin{pmatrix} p_1 & p_2 & 0 \\ p_3 & p_4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad P' = \begin{pmatrix} p'_1 & p'_2 & 0 \\ p'_3 & p'_4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (7)$$

Although these conditions ensure that the epipolar lines of  $Z$  and  $Z'$  are parallel, corresponding epipolar lines are for an different scale factor. Thus, additional constraints are needed (see figure 5(b)). When  $P$  and  $P'$  map corresponding epipolar lines to the same scan line the fundamental matrix will be of the form

$$F = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (8)$$

and therefore the following must hold,  $P = P^T U$ ,  $P' = P^T U'$ . These equations contain two independent constraints,  $-p'_1 p_2 = p_1 p'_2$  and  $p'_3 p_4 = p_3 p'_4$ , that will be satisfied when

$$P = \begin{pmatrix} p_1 & p_2 & 0 \\ p_3 & p_4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad P' = \begin{pmatrix} p_1 & -p_2 & 0 \\ p_3 & -p_4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (9)$$

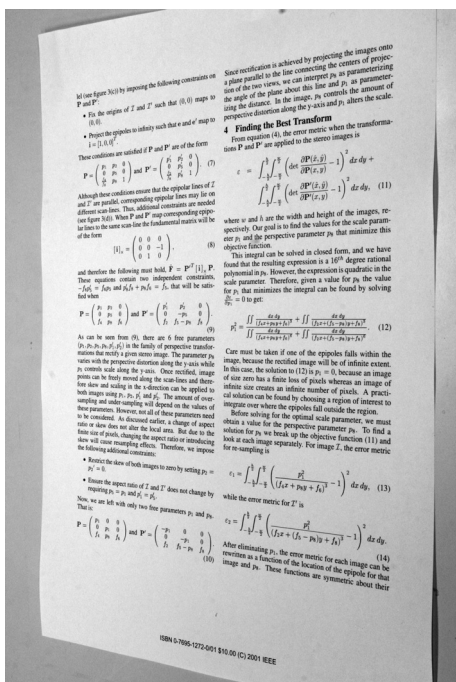
As can be seen from (9), there are 4 free parameters  $(p_1, p_2, p_3, p_4)$  in the family of perspective transformations that satisfy a given stereo image. The parameter  $p_1$  varies with the perspective distortion along the  $y$ -axis while  $p_2, p_3$  and  $p_4$  vary along the  $x$ -axis. Once rectified, image points can be freely moved along the scan lines and therefore along the  $x$ -axis. This amount of correspondence and understanding will depend on the values of these parameters. However, not all of these parameters need to be considered. As discussed earlier, a change of aspect ratio or skew does not alter the local area. But due to the large size of pixels, changing the aspect ratio or introducing skew will cause resampling effects. Therefore, we impose the following additional constraints:

- Rectify the skew of both images to zero by setting  $p_3 = p_4 = 0$ .
- Fix the aspect ratio of  $Z$  and  $Z'$  does not change by imposing  $p_1 = p_2$  and  $p'_1 = p'_2$ .

Now, we are left with only two free parameters  $p_1$  and  $p_3$ . Thus:

$$P = \begin{pmatrix} p_1 & p_1 & 0 \\ p_3 & -p_3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad P' = \begin{pmatrix} p_1 & -p_1 & 0 \\ p_3 & -p_3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (10)$$

(a) データベース中の画像



(b) 検索質問の画像

図 5 文書画像の例

データベースのサイズにあまり依存しないことが分かった。また、検索の精度もほぼ一定となった。

画像を検索質問とした画像検索という枠組みでは本稿のタスクは比較的大規模といえるが、一般の検索の要望を考えた場合には、画像数が十分とは言えない。今後は、さらに画像数を増やすことのできる枠組みを考える必要がある。