

# 大規模特定物体認識における認識率，処理時間，メモリ量のバランスに関する実験的検討\*

野口 和人<sup>†</sup>      黄瀬 浩一<sup>†a)</sup>      岩村 雅一<sup>†b)</sup>

Experimental Evaluation of Balancing the Recognition Rate, Processing Time and Memory Requirement for Large-Scale Recognition of Specific Objects\*

Kazuto NOGUCHI<sup>†</sup>, Koichi KISE<sup>†a)</sup>, and Masakazu IWAMURA<sup>†b)</sup>

あらまし 局所特徴量を用いた特定物体認識において，認識率，処理時間，メモリ量について良いバランスを実現することは重要な問題である．局所特徴量が高い識別性をもつことから，高い認識率を実現することは困難ではないものの，その前提として長い処理時間や大きなメモリ量が要求されるため，大規模化の障害となっている．本論文では，認識率にあまり影響を及ぼすことなく，処理時間やメモリ量を削減する手法について述べる．処理時間の削減については，近似最近傍探索が有効であることを述べるとともに，従来法である ANN と比べて，ハッシュを用いた，より認識に有効な手法を提案する．メモリ量については，局所特徴量を表す特徴ベクトルに対してスカラ量子化を適用することにより，簡単に削減可能であることを示す．10 万画像を用いた認識実験の結果，各次元の表現に用いるビット数を 16 ビットから 2 ビットに減少させても認識率に影響をほとんど及ぼさないこと，提案手法によって，認識率 98.1%，処理時間 119.7ms を実現可能であることが分かった．また，1 万画像を用いた ANN との比較では，同程度の認識率を実現するための処理時間，メモリ量が，それぞれ 1/4，1/3 であることも確認した．

キーワード 物体認識，局所特徴量，近似最近傍探索，ハッシュ表，スカラ量子化

## 1. ま え が き

近年，物体認識が注目を集めている [1]．これは，物体認識に有効な特徴量として SIFT (Scale-Invariant Feature Transform) [2] などの局所特徴量 (local feature) が利用可能となったことによる．現在，局所特徴量を用いて研究されている物体認識は，大きく，一般物体認識 (generic object recognition) と特定物体認識 (specific object recognition) に分類できる．一般物体認識とは，自動車や椅子など，画像に写った物体のクラスを識別するタスクである．一方，特定物体認識は，ある特定の物体 (自動車でいえば特定の車種) が写っているかどうかを識別するというタスクである．本論文は，このうち，後者の特定物体認識に関する．

特定物体認識の中でも，ポスターや写真などの平面物体を対象とした認識は，例えば，バーコード認識の代替という実用的な意味をもつ．対象そのものが高い精度で認識できれば，対象にバーコードを貼付せずに同様のサービスが提供可能となる．これにより，物体の外観がバーコードで悪化することを防げたり，バーコード貼付になじまない対象 (美術品や他人の所有物など) についても，サービスを提供できる．

このような平面物体を対象とした特定物体認識は，局所特徴量のもつ優れた識別性により，高い認識率を達成することができる．識別性とは，「類似した局所特徴量は同じ対象から得られたものであり，そうでなければ違う対象から得られたものである」という性質である．したがって，画像から得た局所特徴量を識別対象物体から得た局所特徴量と照合し，当該物体に投票することにより，簡単に認識が実現できる [3]．

しかしながら，まだ課題も残されている．識別対象となる平面物体の数が大規模 (例えば 10 万物体) になると，抽出される局所特徴量の数が膨大 (10 万物体

<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 Gakuencho, Naka, Sakai-shi, 599-8531 Japan

a) E-mail: kise@cs.osakafu-u.ac.jp

b) E-mail: masa@cs.osakafu-u.ac.jp

\* 本論文は第 12 回画像の認識・理解シンポジウム推薦論文である．

の場合には合計数億個)となる。その結果、(1) 局所特徴量の記憶に膨大なメモリが必要となる、(2) 局所特徴量の照合に長い処理時間が必要となる、というメモリ量、処理時間に関する問題が生じる。

この問題を解決する方策は、既にいくつか提案されている [2], [4], [5]。これらはいずれも木構造を用いて高速化、省メモリ化を図るものである。例えば、Nistérらは、認識対象から得た局所特徴量を vocabulary tree と呼ばれる木構造を用いて量子化し、認識に用いている。このとき、各局所特徴量は、木構造の葉 (visual word [6] に相当) に対応づけられる。visual word の数が局所特徴量の数に比べて十分少ない場合には、処理時間やメモリ量の削減が実現できる。しかし、特定物体の認識率を高めるには、大量の visual word (Nistérらの論文 [5] では 1,000 万以上) が必要と述べられている。visual word の数が大幅に増えれば、処理時間やメモリ量に対する利点は失われる。すなわち、問題が十分解決されているとはいえない。

そこで本論文では、この問題を解決するため、新しい手法を提案する。中心となる考え方は、従来法とは異なる「近似」の導入によって認識率、処理時間、メモリ量の「バランス」をとることである。具体的には、(1) 無駄の削減 (例えば、同じ認識率、処理時間ならメモリ量を少なくする)、(2) バランスの自由な変更 (認識率を落としても処理時間やメモリ量を節約したいといった要望にこたえられること) の二つを目標とする。

物体認識の処理を、上記のように局所特徴量の照合と投票によって実現することを考えてみる。照合結果に多少誤りがあっても、正解の物体に対する票が他の物体に対する票を下回らなければ正しく認識できる。ところが、多くの場合、実際に処理を行うと、1位の得票数 (正解物体に対応) が他の得票数に比べて圧倒的に多いという現象が見受けられる。これは、誤認識を避けるためのマージンを取り過ぎている状態、すなわち認識率、処理時間、メモリ量のバランスが崩れた状態といえる。そこで、認識率の低下というリスクを負うものの、得票数の差が減ることを容認し、その代わりに、処理時間の短縮とメモリ量の削減を試みることを考える。具体的には、照合の精度や局所特徴量の識別性を低下させることによって、これを達成する。

処理時間を短縮する方法としては、近似最近傍探索を導入する。これにより、照合の精度を犠牲にする代わりに処理時間を劇的に短縮できる。このような近似による高速化は、例えば Lowe [2] も導入している。

この手法では Best-Bin-First アルゴリズムを適用し、高速化を達成している。これに対して、本手法では、簡単なハッシュ表を用いる手法 [7] を提案する。また、実験によって、Lowe らが用いた手法の発展形である ANN [8] を用いる場合と比べて高速であることを示す。

メモリ量の削減については以下のとおりである。まず、局所特徴量の表現方法として、visual word のような固定された語彙に変換して用いるのではなく、局所特徴量そのものを用いる。これは、得られた局所特徴量そのものを visual word として用いることといえるため、先に述べた「大量の visual word」という点では最も極端な方法となる。このとき、メモリ量の問題が生じるが、それに対しては、個々の局所特徴量を近似表現して、必要なメモリ量を削減するという方針で臨む。具体的には、局所特徴量を表す特徴ベクトルをスカラ量子化することによって実現する。これは、visual word がベクトル量子化によって得られることとの対比となっている。

また実験により、局所特徴量の各次元を記録するビット数を 16 ビットから 2 ビットに減少させても認識率はほとんど低下しないことを示す。ANN を用いた手法と比べると、提案手法は、同程度の認識率をメモリ使用量 1/3、処理時間 1/4 で達成できることも示す。

## 2. 近似最近傍探索とその従来法

提案手法の詳細について述べる前に、近似最近傍探索とその従来法 ANN についてまとめておく。ANN は、比較実験に用いるものである。

最近傍探索で最も時間がかかるのは距離計算である。最近傍探索の高速化には、個々の距離計算自体を高速に行う方法と、距離計算の対象を効率良く絞り込む方法がある。近似最近傍探索では、主に後者の絞り込みを大胆に行うことで、処理時間を削減する。ところがその代償として、最近傍が距離計算の対象から外れてしまい、求まらない可能性がある。どの程度の近似が適切であるのかは、対象とするタスクに依存するため、近似最近傍探索の手法では、近似の程度を調節するためのパラメータを設けている。以下では、近似最近傍探索の代表的な手法として ANN について述べる。

ANN [8] は、木構造を用いて近似最近傍探索を高速に行う手法である。木のノードは、特徴空間を分割した hyperrectangle に対応しており、葉ノードには特徴ベクトルが対応づけられている。以降、葉ノードに対応する hyperrectangle をセルと呼ぶ。今、 $q$  を検索

質問の特徴ベクトル,  $p_i$  をデータベース中の画像の特徴ベクトルとする. ANN では, まず木構造を用いて  $q$  がどのセル内に存在するかを発見する. 発見したセルに対応する特徴ベクトルを  $p_s$  とすると, 最近傍は  $p_s$  と  $q$  の距離  $r$  を半径とする超球内に存在する. そこで, その超球と重なるセルを順次探索し最近傍を得る. 近似最近傍探索を行う場合, 許容誤差  $\varepsilon$  を用いて定義される半径  $r/(1+\varepsilon)$  の超球を考え, それと交わるセルのみを探索の対象とする.  $\varepsilon$  が大きければ, 最近傍の特徴ベクトルを発見できない可能性は大きくなるが, 対象となるセルの数が減少するため, 探索時間を削減できる.

### 3. 提案手法

#### 3.1 構成

本論文で提案する物体認識は以下のようなものである. 本手法では, 認識処理の前に, 認識対象物体の画像から多数の局所特徴量を求め, 得られた特徴ベクトルを画像 ID とともにデータベースに登録しておく. 検索質問が与えられたとき, 検索質問と同じ物体が写った画像をデータベースから検索する. 検索では, 検索質問の特徴ベクトルに対して, 最近傍となる特徴ベクトルをデータベースから探索する. そして, 最近傍の特徴ベクトルに対応する画像に投票し, 得票数が最大となる画像を認識結果として回答する.

局所特徴量としては, PCA-SIFT [9] を用いる. PCA-SIFT は SIFT の改良手法であり, 36 次元の特徴ベクトルが得られる.

提案手法は, ハッシュとスカラ量子化という二つの方法によって高速化と省メモリ化を図るものである. これらの導入は, 認識率にとって悪影響を及ぼすことが考えられる. そこで本論文では, ハッシュやスカラ量子化のパラメータと, 認識率との関係を, 10 万平面物体という規模のデータを用いて実験的に明らかにする. 以下, 説明の都合上, スカラ量子化から順に述べる.

#### 3.2 スカラ量子化

##### 3.2.1 考え方

1. でも述べたように, 通常, 局所特徴量に対して適用される量子化はベクトル量子化である [6]. 特に一般物体認識においては, ベクトル量子化の適用は基本的な技術として定着している. ところが, 大規模特定物体認識のように, 識別すべき対象が多数となり, そのために扱う局所特徴量の数が膨大となると問題が生じてくる. 具体的には visual word を定めるためのクラ

スタリング処理が問題となる. 数千万を超えるような膨大な数の高次元ベクトルに, クラスタリング処理を施すことは容易ではない.

この問題を解決する方策には, サンプリングを行って少数のベクトルのみを対象としてクラスタリング処理を施すことや, vocabulary tree のように階層的にクラスタリング処理を施すことなどが考えられる. いずれにせよ, クラスタリングの結果として得られる visual word の最適性については, あきらめざるを得ない.

visual word を用いるもう一つの問題点は, 先にも述べたとおり, 新たな認識対象の扱いが簡単ではないことである. visual word を定めるためには, 局所特徴量の分布を仮定しなければならず, 大幅に傾向の異なる局所特徴量が新しい認識対象から得られた場合, visual word による表現が適切ではなく, 認識率に悪影響を及ぼすことが考えられる.

更に, 一般物体認識の場合と異なり, 大規模特定物体認識の場合には, 数百から数千万個という非常に多くの visual word を用いることが, 認識性能を高める上で必要となってくる. 認識対象の規模にもよるが, これは, 認識対象の画像から得られる数個から数十個の局所特徴量に対して一つの visual word を定めていることに相当するので, メモリ削減の効果も限定的になってくると考えられる.

そこで本手法では, 局所特徴量の分布にあまり依存することなく特徴空間をカバーし, メモリ削減効果もねらえる手法として, スカラ量子化を導入する. 特徴ベクトルの次元ごとに量子化を行うため, ベクトル量子化に比べて必要なビット数は増えるが, 局所特徴量の分布に大きく影響を受けない量子化が可能となる.

##### 3.2.2 方法

提案手法で用いるスカラ量子化について説明する. 各次元 2 bit で量子化する場合を図 1 に示す.  $f(x)$  は特徴ベクトルのある次元の頻度分布である.  $t_i$  は量子

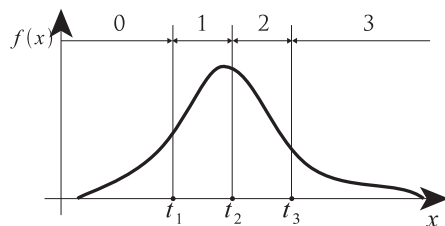


図 1 スカラ量子化  
Fig. 1 Scalar quantization.

化のしきい値であり、例えば  $-\infty$  から  $t_1$  の範囲のものは 0 に量子化する。  $t_i$  については、量子化によって得られる値（上記では 0~3）の出現確率が均等になるように定める。

### 3.3 ハッシュ

#### 3.3.1 考え方

物体認識にハッシュを用いる手法は、80 年代後半に提案された Geometric hashing [10] や、それに続く発展形 [11] に既に現れており、木構造を用いる手法と並んで有効であることが知られている。Indyk らによって LSH (Locality Sensitive Hashing) (例えば [12]) が提案されて以来、理論的な整備も進み、物体認識や画像検索などへの応用も加速している [13]。

本論文で提案する手法は、ハッシュや LSH の単なる適用ではなく、次の問題点に対する工夫を伴うものである。ハッシュを用いて近似最近傍探索を行う場合、中心的な課題は、いかに探索漏れを防ぐかにある。ハッシュでは、入力の実数値ベクトルを、ハッシュ関数を用いて整数のインデックスに変換する。このとき、撮影条件などの影響でベクトルの値が変動すると、同じインデックスが得られないという問題が生じる。

この問題は、様々なハッシュ関数を用いて解決できる。これは、あるハッシュ関数で検索に失敗しても、他でカバーできるという考えである。このような考えから、LSH では、複数のハッシュ関数とそれに対応したハッシュ表を用い、個々のベクトルをすべてのハッシュ表に登録することで探索漏れの確率を減らしている。この方策は小規模なデータに対しては有効である。しかし、我々のタスクのように大規模なデータ（数億ベクトル）の場合には、(1) 十分な性能を得るには多数のハッシュ表が必要であり、(2) 各ハッシュ表は大量のベクトルを区別するために大きくならざるを得ず、(3) 一つのベクトルをハッシュ表の数だけ重複して登録する必要があることから、メモリ量の問題が生じる。

この LSH の問題を解決する一方策は、ハッシュ表を一つに制限することであろう。この条件下で探索漏れを防ぐ方法には、次の 2 通りが考えられる。

- 多重登録：複数のハッシュ関数を用いて登録する点は LSH と同じであるが、個別のハッシュ表ではなく同一のハッシュ表に登録する手法。

- 多重探索：登録時ではなく探索時に複数のハッシュ関数を適用し、得られたインデックスを用いて複数回探索する手法。

前者は LSH の (3) の問題が同様に生じるため望ま

くない。そこで本手法では後者を採用する。

#### 3.3.2 ハッシュ関数

まず基本となるハッシュ関数について説明する。PCA-SIFT によって得られる 36 次元の実数値ベクトル  $x$  は、主成分分析の結果であり、低次の方が固有値が大きいという性質がある。そこで、 $x$  の第 1 次元から第  $d$  次元 ( $d \leq 36$ ) までをとり、 $\hat{x} = (x_1, x_2, \dots, x_d)$  とする。次に、

$$u_j = \begin{cases} 1 & \text{if } x_j - \mu_j \geq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

を用いて次元ごとに 2 値化を行いビットベクトル  $u = (u_1, \dots, u_d)$  を作成する<sup>(注1)</sup>。ここで  $\mu_j$  は、認識対象となる画像すべての  $x_j$  の平均値である。そして、

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

によってハッシュ値を求める。ここで  $\bmod$  は剰余演算、 $H_{\text{size}}$  は、ハッシュ表のサイズである。

#### 3.3.3 データ登録

登録処理の流れを図 2 の上部に示す。まず特徴ベクトルをハッシュ表に登録するためには、前述のハッシュ関数によりハッシュ値を求める必要がある。ハッシュ表には、画像 ID とともに量子化によってデータ量を削減した特徴ベクトルを登録する。登録時に衝突が生じた場合は、複数の特徴ベクトルをリストとして登録する。このとき、リストが長くなりすぎると、検

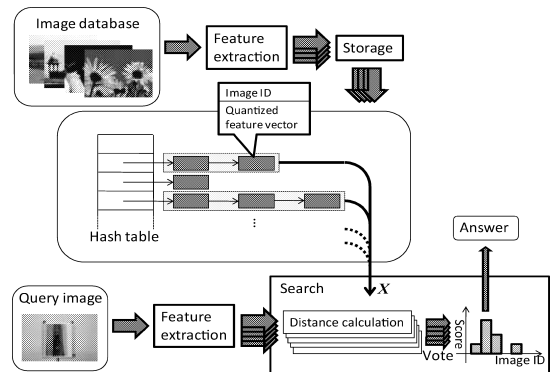


図 2 処理の流れ

Fig. 2 Processing flowchart.

(注1): ここで行う量子化はハッシュ値を求めるためのものであり、3.2.2 で説明したスカラー量子化とは異なる。そのため、3.2.2 の量子化のパラメータを変化させても、ハッシュ値は変化しない。

索の際に距離計算のコストがかかりすぎるという問題が生じる．そこで本手法では, リスト長  $n$  に対する上限  $c$  を設け,

$$n > c \quad (3)$$

を満たすとリスト全体をハッシュ表から削除する．同じハッシュ値をもつ特徴ベクトルが多いということは, その特徴ベクトルが画像の識別にあまり寄与しないことを意味する．したがって, 削除しても影響は比較的小さいと考えられる．

以上の処理を, データベースに登録するすべての特徴ベクトルに対して施すことにより, データの登録は完了する．

### 3.3.4 検 索

検索処理の流れを図 2 の下部に示す．本手法では, 検索質問の画像から得た各特徴ベクトル  $q$  に対して, ハッシュ関数を適用しハッシュ表から特徴ベクトルを検索する．ここで得られた特徴ベクトルの集合を  $X$  とする．次に,  $q$  を量子化したベクトルと,  $X$  に含まれるベクトルとのユークリッド距離を計算し, 最近傍となる特徴ベクトル  $x_*$  を求める．

そして,  $x_*$  に対応する画像 ID に投票する．最近傍となる特徴ベクトルが複数ある場合には, それらすべてに対して投票処理を施す．検索質問のすべての特徴ベクトルに対してこの処理を行い, 最終的に最も得票数の多いものを回答とする．

この処理において, 最も重要なステップは, いかに  $q$  に対する特徴ベクトルの集合  $X$  を検索するかにある．最も単純な手法は, 登録時と同様に  $q$  に対してもビットベクトルを求め, ハッシュ関数によって同じハッシュ値をもつ特徴ベクトルを求めることである．ところが, このような処理では, 距離の計算回数は十分削減できるものの, 次の理由によって十分な認識率を得ることができない．それは, 特徴ベクトルの各次元の値が撮影条件によって変動するためである．もし, しきい値を超えるような変動があると, ビットベクトルが異なるものとなり, もはや対応する特徴ベクトルを得ることができなくなる．

そこで本手法では, ビットベクトルを操作することによって, 変動を考慮した探索を行う．具体的には, 値の変動幅  $e$  をパラメータとして, 変動への次の対処を施す． $q = (q_1, \dots, q_d)$  とするとき,

$$|q_j - \mu_j| \leq e \quad (4)$$

を満たす次元  $j$  に対しては,  $u_j$  だけではなく  $u'_j = (u_j + 1) \bmod 2$  ( $0$  ならば  $1$ ,  $1$  ならば  $0$ ) も用いて, 特徴ベクトルを検索する．

ただし, このような操作を制限なく導入すると, 膨大な計算時間が必要となってしまう．そこで, 操作する次元数を  $b$  に制限する．これは,  $2^b$  通りのビットベクトルを用いてハッシュ表にアクセスすることを意味し, 3.3.1 で述べた, 複数 (この場合は  $2^b$  個) のハッシュ関数を用いることに相当する． $|q_j - \mu_j| \leq e$  を満たす次元の数が  $b$  を上回るときには, 対象とする次元を, 高次の方から  $b$  個とする．

## 4. 実 験

### 4.1 実験条件

実験では, 近似最近傍探索の従来法である ANN<sup>(注2)</sup>を用いた認識手法と提案手法を比較した．

実験には以下に述べる画像データベース, 検索質問画像を用いた．局所特徴量としては, PCA-SIFT<sup>(注3)</sup>を用いた．ハッシュ表のサイズは  $H_{\text{size}} = 2^d$  とした．以下に示す処理時間は, 検索質問の画像 1 枚当りの認識に要した時間を表す．ただし, 特徴ベクトルの抽出に必要な時間は含めていない．使用計算機は, CPU が AMD Opteron 2.8 GHz, メモリ 32 GByte のものである．

#### 4.1.1 画像データベース

実験に用いた画像について説明する．まず, 収集方法の異なる A, B, C の 3 種類のデータセットを準備した．A は, Google のイメージ検索を用いて収集した 3,100 枚の画像である．検索キーワードとしては, “ポスター”, “雑誌”, “表紙” などを用いた．図 3 (a) に例を示す．B は PCA-SIFT のサイトで公開されていた画像であり, 画像数は 18,500 枚である．このデータは主に自然写真や人物の写真などで構成されている．

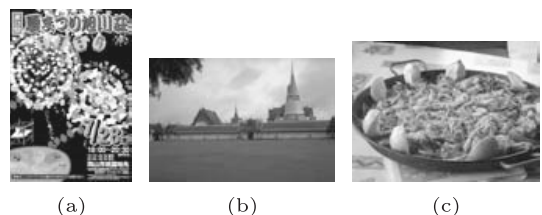


図 3 登録画像の例

Fig. 3 Examples of images in a database.

(注2): ANN としては <http://www.cs.umd.edu/~mount/ANN/> で提供されているソースコードを用いた．

(注3): <http://www.cs.cmu.edu/~yke/pcasift/> で提供されている．

図 3 (b) に例を示す．C は、写真共有サイトの flickr において “animal”，“birthday”，“food”，“japan” などのタグにより収集した 78,400 枚の画像からなる．主に図 3 (c) に示すような物体や自然の写真，人物の写真などを含む．なお、収集の際には、600 × 600 pixel 以下のサイズの画像は除外し、画像の長辺が 640 pixel 以下になるように縮小した．また、特徴ベクトルが 100 個以下の画像も除外した．画像の 1 辺の長さの平均は A, B, C それぞれ 498, 612, 554 pixel であった．

次に、A, B, C の画像を用いて、表 1 に示した画像数からなる 5 種類のデータベースを作成し実験に用いた．ここで、大きいデータベースは、小さいデータベースをその一部として含む．

また、特徴ベクトルの個数は画像 1 枚当たり平均 2,000 個であった．実験では、どの枚数のデータベースを用いる場合でも、1 万枚の画像データベースから得られた  $f(x)$  を量子化に用いた．

4.1.2 検索質問画像

検索質問としては、データセット A, B, C のそれぞれから 100, 200, 200 枚の合計 500 枚を無作為に選択した．次に、これらを A4 の用紙に印刷し、カメラを用いて撮影した．得られた画像の例を図 4 に示

表 1 データベースに含まれる画像数  
Table 1 The number of images in the databases.

登録画像数	1,000	5,000	10,000	50,000	100,000
内訳	A	334	1,667	3,100	3,100
	B	333	1,667	3,450	18,500
	C	333	1,666	3,450	28,400

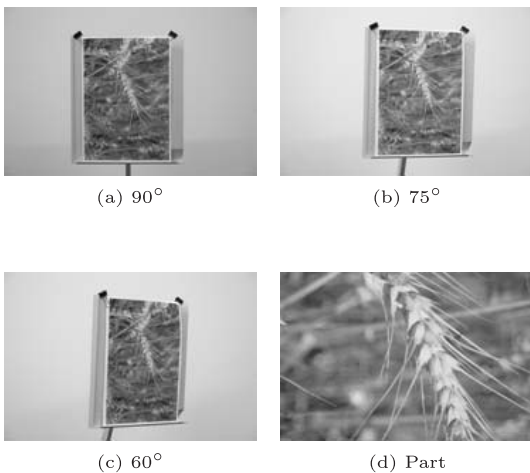


図 4 検索質問の例  
Fig. 4 Examples of query images.

す．図に示すとおり、紙面全体が写る配置で、紙面に対するカメラの光軸の角度  $\theta$  を  $90^\circ, 75^\circ, 60^\circ$  に変化させた．また、角度を  $90^\circ$  として紙面の一部分を撮影した．その結果、1 枚の紙面に対して、合計 4 通りの画像を得た．更に、撮影した画像を  $512 \times 341$  pixel に縮小し、PCA-SIFT により特徴ベクトルを求めた．その結果、画像 1 枚当たり平均 612 個の特徴ベクトルが得られた．

4.2 提案手法のパラメータと認識率・処理時間

実験の結果について述べる前に、提案手法の六つのパラメータについて確認しておく．一つ目のパラメータは、次元当りの量子化 bit 数である．残りはハッシュのパラメータで、ハッシュ関数に使う次元数  $d$  (式 (2)), ハッシュ表のサイズ  $H_{size}$  (式 (2)), ハッシュ表のリスト長の上限  $c$  (式 (3)), ビットベクトルで反転の操作を行う次元数  $b$ , 値の変動幅  $e$  (式 (4)) の五つである．

まず量子化を行う前に、PCA-SIFT で得られたそのままのデータ (次元当り 16 bit で表現) を用い、ハッシュのパラメータ  $b, c, d, e$  と認識率、処理時間の関係について調べた．なお、すべての実験でハッシュ表のサイズ  $H_{size} = 2^d$  とした．

まず、リスト長の上限  $c$ 、ハッシュ関数に使う次元数  $d$  と認識率、処理時間の関係について述べる．画像データベースは 1 万枚とした． $e = 200, b = 10, d = 24, 26, 28$  とし、 $c$  を変化させた結果を図 5 に示す．認識率は  $c$  を小さくしすぎると低下した．これは、認識に寄与していたものも削除したためと考えら

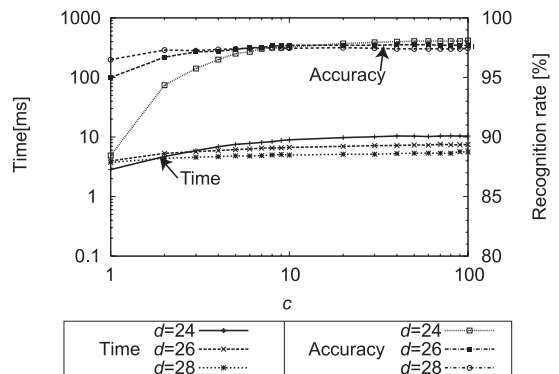


図 5 リスト長の上限  $c$  と認識率、処理時間の関係  
Fig. 5 The relationship among the upper limit of the list length  $c$ , the number of dimension  $d$  for the hash function, the recognition rate and processing time.

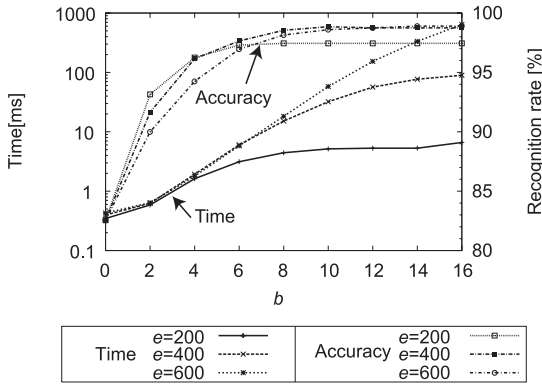


図 6 ビットベクトルを操作する次元数  $b$ , 値の変動幅  $e$  と認識率, 処理時間の関係

Fig. 6 The relationship among the number  $b$  of dimensions of flipping the bit vector, the value range  $e$ , the recognition rate and processing time.

れる。一方,  $c$  を 10 以上に増加させると, 計算時間は多少増加するものの, 認識率が改善することはほとんどなかった。このことから, リスト長の上限  $c$  は一定以上の値であればよいといえる。

次に, ビットベクトルで反転の操作を行う次元数  $b$ , 値の変動幅  $e$  と認識率, 処理時間の関係について述べる。 $d = 28, c = 10$  とした上で, 値の変動幅  $e$  と操作する次元数  $b$  を変化させた結果を図 6 に示す。 $b$  を増加させると, 処理時間は増加するものの, 認識率は改善する。注目すべき点は,  $b = 0$  の場合, すなわち基本となるハッシュ関数のみを用い, ビットベクトルを操作しない場合には, 認識率が極端に低いことである。 $b$  を増加させると, 処理時間は指数関数的に増えるものの, 認識率も劇的に改善する。認識率の改善は  $b = 10$  程度で落ち着くので, このときの処理時間が許容できればよい。以後の実験で用いる  $e = 400$  のときには, 30 ms 程度となっている。

### 4.3 メモリ量と認識率

次に, 量子化によりデータ量を削減すると, どの程度認識率に影響がでるかを調べた。結果を図 7 に示す。横軸は特徴ベクトルの各次元の量子化ビット数を表している。例えば, 2 bit の場合には, PCA-SIFT の特徴ベクトルが 36 次元なので, 1 ベクトル当り  $2 \text{ bit} \times 36 = 72 \text{ bit}$  必要となる。16 bit の場合は, 先ほどの実験と同様に, 量子化を行わずもとのデータをそのまま用いることを意味する。また, 0 bit の場合には, 距離計算を行うための特徴ベクトルが存在しないため, ハッシュ表から得られた特徴ベクトルの集合  $X$

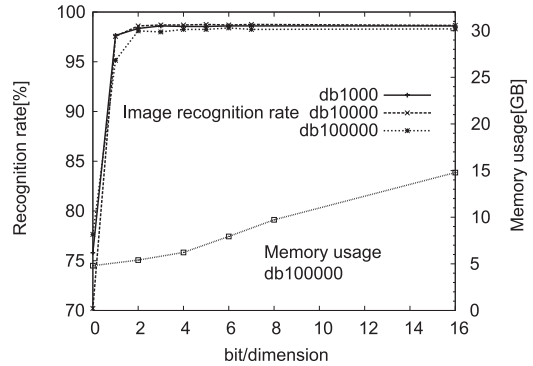


図 7 特徴ベクトルの容量と認識率, メモリ使用量の関係  
Fig. 7 Relationship among bit/dimension, the recognition rate and memory size.

のすべてに対して投票を行うことを意味する。

登録画像数を 1,000, 1 万, 10 万枚, パラメータを先ほどの実験で良い結果の得られた  $b = 10, c = 10, d = 28, e = 400$  として実験した。また, メモリ使用量は 10 万枚のデータベースについてのみ調べた。

図 7 のグラフから, 2 bit まで削減しても, 認識率に大きな影響は見られないことが分かる。1 bit の場合には登録画像数が多いほど, 認識率の低下が顕著となった。これは, 登録されている特徴ベクトルの数が増えるほど, 識別性の不足が問題となるためと思われる。0 bit の場合には認識率が大きく低下した。

以上より, 高い認識率を得るための表現としては, 各次元 16 bit は不要であり, 各次元 2 bit で十分であることが分かる。各次元を 16 bit で表現した場合と比べると, 特徴ベクトル単体で  $1/8$  の容量であり, ハッシュ表などを含むメモリ使用量全体では, 図 7 から約  $1/3$  と大幅な圧縮が可能であることが分かる。

### 4.4 2 bit と 16 bit で量子化する場合の認識率・処理時間

次に, 提案手法の 2 bit と 16 bit の場合について登録画像数と認識率と処理時間の関係を比較した。登録画像数を 5,000 枚から 10 万枚まで変化させ, 実験を行った。パラメータは先ほどと同様に  $b = 10, c = 10, d = 28, e = 400$  を用いた。登録画像数を変化させた結果を図 8 に示す。16 bit の場合と比べて, 2 bit の場合, 認識率は約 0.3% 低下した。処理時間については, ほとんど変化が見られなかった。登録画像数 10 万枚における認識率と処理時間は, 16 bit の場合 98.35%, 107.4 ms であり, 2 bit の場合 98.1%, 119.7 ms であった。以上より, 2 bit の表現は, 登録画像数が増加して

表 2 各手法の認識率 [%] と処理時間 [ms]  
Table 2 Recogniton rate [%] and processing time [ms] of each method.

手法	パラメータ	60 度		75 度		90 度		一部		平均	
		精度	時間	精度	時間	精度	時間	精度	時間	精度	時間
ANN 16 bit	$\epsilon = 3$	98.8	973.7	99.0	1004.0	99.0	1005.5	99.4	3165.4	99.1	1537.2
	$\epsilon = 6$	98.4	147.3	98.8	155.6	98.8	155.3	99.0	468.2	98.8	231.6
	$\epsilon = 20$	96.0	15.9	97.8	18.3	97.8	18.4	98.0	50.8	97.4	25.9
ANN 2 bit	$\epsilon = 3$	98.6	562.2	99.0	578.0	98.8	576.2	98.8	1835.6	98.8	888.0
	$\epsilon = 6$	96.6	28.9	98.4	32.5	98.4	32.4	97.6	92.4	97.8	46.5
	$\epsilon = 20$	69.0	2.4	92.4	3.0	93.8	3.1	93.6	8.6	87.2	4.3
提案手法 16 bit	$b = 10, c = 10, d = 24, e = 600$	98.8	111.7	99.0	136.2	98.6	139.0	99.0	373.9	98.9	190.2
	$b = 12, c = 10, d = 28, e = 400$	98.2	33.3	98.8	40.6	98.8	41.5	99.2	111.2	98.8	56.6
	$b = 6, c = 10, d = 28, e = 200$	95.0	1.7	98.4	2.2	98.0	2.3	97.8	6.4	97.3	3.1
提案手法 2 bit	$b = 10, c = 10, d = 24, e = 600$	98.2	135.9	98.8	165.8	98.8	169.2	98.8	453.8	98.7	231.2
	$b = 12, c = 10, d = 28, e = 400$	98.2	26.3	98.8	32.0	98.6	32.8	99.0	87.6	98.7	44.7
	$b = 6, c = 10, d = 28, e = 200$	95.4	1.7	98.2	2.2	98.0	2.3	97.4	6.5	97.3	3.2

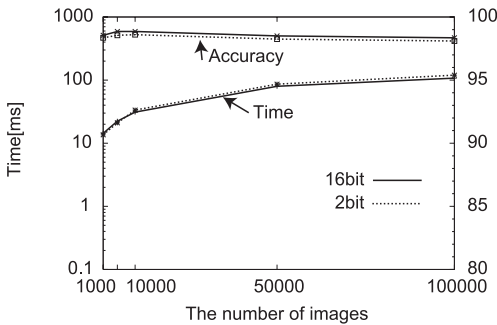


図 8 登録画像数と認識率、処理時間の関係

Fig. 8 Relationship among the number of images in the DB, the recognition rate and processing time.

も安定して有効であることが分かった。

更に、誤認識を起こした検索質問について調べた。その結果、2 bit と 16 bit の場合で誤認識の傾向にはほとんど差がないこと、誤認識となる検索質問は特定の登録画像に集中していることが分かった。これらの画像が誤認識を起こしやすい理由は、エッジがぼやけており、検索質問の特徴点数が少ないためと考えられる。

#### 4.5 ANN を用いた手法との比較

近似最近傍探索の従来法である ANN を用いた認識手法と、提案手法を比較した。1 万画像のデータベースでパラメータを様々に変え、認識率、処理時間について調べた。横軸に認識率、縦軸に処理時間を描いたグラフを図 9 に、各種パラメータの代表的な値を用いた認識率と処理時間を表 2 に示す。

ANN で許容誤差  $\epsilon$  を 2 から 1,000 まで変化させたものを線で描き、評価の基準とした。右にプロットされているものほど、認識率が高く、下にプロットされているものほど、処理時間が短い。そのため、右下に

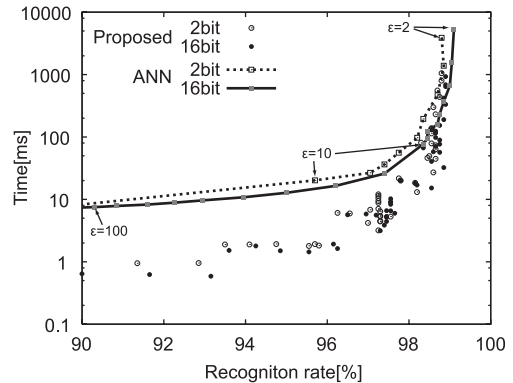


図 9 ANN を用いた認識手法との比較

Fig. 9 Comparison to a recognition method with ANN.

プロットされているものほど、優れているといえる。このグラフから次のことが分かる。

- ANN を用いた手法と提案手法のいずれにおいても、16 bit の表現と 2 bit の表現で大差のない性能を得ることができる。
- 最も高い認識率は ANN を用いた手法によって得られる。ただし、1 秒を超える処理時間が必要である。
- 99%未満の認識率で比べると、ANN と同じ認識率を達成するために必要な処理時間が、提案手法は  $1/4 \sim 1/10$  となることが多く、おおむね優れているといえる。

以上から、提案手法は ANN を用いた手法と比べてより良く認識率、処理時間、メモリ量のバランスをとることが可能であるといえる。



## 5. むすび

大規模な特定物体認識では認識率, 処理時間とメモリ量のバランスを考慮することが重要であるとの観点から, 本論文では, ハッシュとスカラ量子化を用いた特定物体認識法を提案した. 1万枚の画像データベースを用いた実験では, 各次元 2 bit で量子化した場合に, 近似最近傍探索の従来法である ANN を用いた手法と比べ, メモリ使用量を 1/3, 処理時間を 1/4 にすることができた. また, 画像データベースを 10 万画像に増やしたとき, 認識率 98.1%, 処理時間 119.7 ms (局所特徴量抽出時間を除く) を得ることができた.

今後の課題としては, ベクトル量子化を用いた手法との比較, 更なる大規模データでの実験, 特徴点を得にくい画像への対処などが挙げられる.

謝辞 本研究の一部は, 科学研究費補助金 (基盤研究 (B) 19300062) の補助による.

## 文 献

- [1] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, eds., *Toward Category-Level Object Recognition*, Springer, 2006.
- [2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol.60, no.2, pp.91–110, 2004.
- [3] 黄瀬浩一, 岩村雅一, "3日で作る高速特定物体認識システム" 情報処理, vol.49, no.9, pp.1082–1089, Sept. 2008.
- [4] S. Obdržálek and J. Matas, "Sub-linear indexing for large scale object recognition," *Proc. BMVC2005*, vol.1, pp.1–10, 2005.
- [5] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," *Proc. CVPR2006*, pp.775–781, 2006.
- [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] 野口和人, 中居友弘, 黄瀬浩一, 岩村雅一, "特徴ベクトルの近傍探索と物体認識の効率に関する実験的検討" 信学技報, PRMU-229, Sept. 2006.
- [8] S. Arya, D.M. Mount, R. Silverman, and A.Y. Wu, "An optimal algorithm for approximate nearest neighbor searching," *J. ACM*, vol.45, no.6, pp.891–923, 1998.
- [9] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," *Proc. CVPR2004*, vol.2, pp.506–513, 2004.
- [10] H. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE Comput. Sci. Eng.*, vol.4, no.4, pp.10–21, 1997.
- [11] A. Califano and R. Mohan, "Multidimensional indexing for recognizing visual shapes," *IEEE Trans.*

*Pattern Anal. Mach. Intell.*, vol.16, no.4, pp.373–392, 1994.

- [12] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," *Proc. 20th Annual Symposium on Computational Geometry*, pp.253–262, 2004.
  - [13] G. Shakhnarovich, T. Darrell, and P. Indyk, eds., *Nearest-neighbor methods in learning and vision*, MIT Press, 2005.
- (平成 20 年 10 月 14 日受付, 21 年 2 月 25 日再受付)



野口 和人 (学生員)

平 19 阪府大・工・情報卒. 現在, 同大大学院博士前期課程に在籍. 画像認識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, 同大大学院工学研究科教授. 博士 (工学). その間, 平 12–13 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, 画像認識などの研究に従事. 平 18 本会論文賞, 平 19 IAPR/ICDAR Best Paper Award 各受賞. 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. 現在助教. 博士 (工学). パターン認識, 画像認識, 情報検索などの研究に従事. 平 18 本会論文賞, 平 19 IAPR/ICDAR The Best Paper Award 各受賞. IEEE, 情報処理学会, Pattern Recognition Society 各会員.