

# 局所特微量の多数の部分空間を用いた 3 次元物体認識と その高速化・高精度化

柏木 隆宏<sup>†</sup> 黄瀬 浩一<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

E-mail: <sup>†</sup>kashiwagi@m.cs.osakafu-u.ac.jp, <sup>†</sup>kise@cs.osakafu-u.ac.jp

あらまし 莫大な数の部分空間を用いて 3 次元物体を高速に認識する手法を提案する．通常，データベース中の部分空間とクエリとなる特微量の類似度は，その部分空間へ射影した際の射影長を計算することで求められる．そのため，データベース中の部分空間の数が莫大になると，全ての部分空間との類似度を計算し，物体を認識するために，莫大な処理時間が必要となる．そこで我々は、莫大な数の部分空間のデータベースに対して，近似的に探索を行い，高速に物体を認識する手法を提案する．また，近似探索を行った場合でも認識率を維持するため，近似を用いて高速に部分空間の絞り込みを行い，絞り込んだ部分空間において正確な類似度の比較を行う手法も提案する（2 段階処理）．更に，この手法を相互部分空間法にも適用し，3 次元物体の高速で高精度な認識を実現する．実験を行った結果，近似を用いて類似度の高い部分空間を求めて物体を認識した場合，近似を用いない場合に比べて処理時間を 300 分の 1 にすることができた．また，近似を用いて相互部分空間法を行うことで，近似を用いた部分空間法と比べて，認識率が 60% 以上向上した．

キーワード 部分空間法，局所特微量，3 次元物体認識，高速化，近似最近傍探索

## 1. はじめに

近年，デジタルカメラなどで撮影した画像から，コンピュータ処理によって 3 次元物体を認識する手法が多く提案されている [1] [2]．その中でも，部分空間法を用いた手法は，その部分空間上に物体の変化を表すことができ，向きの推定などにも有効であるなどの様々な利点があるため，これまでも多くの研究がなされている [3] [4] [5]．しかし，部分空間法を用いた多くの手法が，画像全体から大域特微量を抽出して用いているため，オクルージョンや照明変化に弱いという問題がある．

大域特微量の問題を解決するために，SIFT [6] などの局所特微量を用いた手法がある．局所特微量とは，1 枚の画像の様々な局所領域から特微量を抽出したもので，画像中に物体の一部しか写っていないような場合や一部が隠れている場合，また回転や照明変化を伴う場合に対しても有効である．しかし，1 枚の画像から数百から数千個の特微量が得られるため，データベースに登録する物体数が増えるほど，その特微量の数は膨大なものとなる．そのため，メモリ使用量が膨大になるという問題や，莫大な処理時間が必要となるという問題が生じる．

我々は以前，局所特微量を用いた物体認識におけるメモリ削減の手法として，部分空間を用いる手法を提案した [7]．この手法では，物体を回転させて連続的に撮影し，その物体の同じ領域から抽出した局所特微量の集合から部分空間を作成しデータベースに登録することで，メモリ使用量を削減する．しかし，認識の際に，クエリとして与えられた特微量をデータベース中の部分空間全てに射影して類似度を求めるため，その処理には

莫大な時間が必要である．

本稿では，莫大な数の部分空間に対して，近似的に最も類似度の高い部分空間を探索することで，高速に物体を認識する手法を提案する．また，相互部分空間法においても，この提案手法を応用して高速に部分空間同士の類似度を比較し，物体の認識を行う．実験では，1002 物体の画像データセットから計約 150 万個の部分空間を作成し，それらを用いて物体の認識を行った．その結果，近似を用いない場合に比べて処理時間を 300 分の 1 にすることができた．また，近似を用いて相互部分空間法を行うことで，近似を用いた部分空間法と比べて，認識率が 60% 以上向上させることができた．

## 2. 関連手法

### 2.1 3 次元物体認識

ここでまず，局所特微量を用いた物体認識手法について述べる．最も単純な認識手法は，次の通りである．まず，あらかじめ物体を様々な角度から撮影し，それらの画像から局所特微量を抽出し，データベースに登録する．そして，クエリ用画像から抽出した各局所特微量とデータベース中のそれとを比較し，最近傍となる特微量をもつ物体 ID に投票を行う．最終的に得票数が最大となる物体を認識結果として出力する．

局所特微量を用いた 3 次元物体認識における初期の手法として，Lowe らによって *view clustering* [1] が提案されている．これは，物体を様々な角度から撮影した画像をデータベースに登録する際に，似ている画像同士をクラスタリングすることで，登録する画像数を減らすというものである．また，Kim らに

よって提案された手法 [8] では、更にデータベースを小さくするために、異なる物体から抽出した局所特徴量であっても、類似していれば共有して登録している。

3次元物体認識に局所特徴量を用いる場合、あらかじめ様々な視点から撮影した画像から局所特徴量を抽出してデータベースに登録するため、登録する物体数が増えれば増えるほど、そのメモリ使用量は膨大なものとなる。そのため、いかにデータベースに登録する特徴数を減らすか、いかにデータベースのメモリ使用量を削減するかが大きな課題となる。

## 2.2 部分空間法

3次元物体認識におけるデータベースのメモリ使用量を削減する手法として、部分空間を用いた手法がある [9]。この手法では、それぞれの物体を様々な視点から撮影し、それら画像から得られる特徴の集合を主成分分析することで、部分空間として各物体を表現し、メモリ使用量を削減している。部分空間法では、クエリ画像から得られた特徴量を、各物体の部分空間に射影することで、その特徴量と部分空間の類似度を求め、その類似度を用いてクエリ画像を認識する。

この部分空間法は、これまでに多くの手法に拡張されてきた。その代表的なものの一つに、相互部分空間法 [5] と呼ばれるものがある。相互部分空間法では、データベースとクエリとも部分空間を用いて、それらの類似度によって物体を認識する。まず、クエリとして物体を様々な視点から撮影した複数の画像を与え、それらからデータベースと同様に部分空間を作成する。そして、その部分空間とデータベース中の部分空間との正準角によって類似度を求め、物体を認識する。そうすることで、クエリとして特徴量を用いる場合より、高精度な認識を行うことができる。

部分空間法に関連するほとんどの手法 [3] [10] が、画像全体から特徴量を抽出し、部分空間を作成している。そのため、データベースに登録する部分空間の数は比較的少ない。一方、我々は局所特徴量ごとに部分空間を作成するため、データベースに登録する部分空間の数は膨大なものとなる。そこで、そのような莫大な数の部分空間を扱うために、類似度の高い部分空間を高速に探索する手法の提案が必要となる。

## 2.3 近似探索

特徴量同士の比較においては、ANN [11] や LSH [12] のように近似を用いて高速に最近傍を探索する手法が多く提案されている。しかし、これらの手法はユークリッド距離における最近傍を探索するものであり、部分空間の探索に直接的に用いることはできない。

最近、Basri らによって *approximate nearest subspace search* [13] と呼ばれる部分空間を近似的に探索する手法が提案された。この手法では、最も類似度の高い部分空間を求める問題を、ベクトル同士の近似最近傍探索の問題へと変換している。しかしながら、[13] でも述べられているように、この手法に必要なメモリ量は莫大で、 $O(d^2)$  ( $d$  はベクトルの次元数) となる。そのため、 $d$  が大きくなればなるほど、必要なメモリ量は増加してしまうという問題があり、容易に用いることはできない。

本稿では、*approximate nearest subspace search* より単純で、

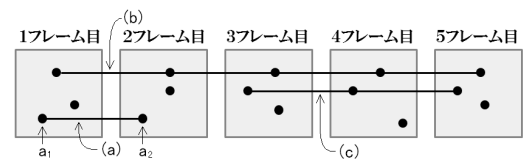


図 1 対応づけられた特徴量の軌跡の例

部分空間の数が莫大な場合においても、高速な処理を行うことができる手法を提案する。本提案手法は、データベース中の莫大な部分空間の 1 次元しか用いないことで、クエリから得られた各特徴量と類似度の高い部分空間を近似最近傍探索によって探索し、高速に 3 次元物体の認識を行う。部分空間の次元数を 1 次元にすることで、情報の欠損は多くなり、部分空間の識別精度が低下するという問題があるが、投票処理を行うことでその問題を解決することができる。

## 3. 提案手法

本手法では、物体を回転させて連続的に撮影し、物体の同じ領域から抽出した局所特徴量から、部分空間を作成する [7]。そうすることで、物体が回転したときの局所特徴量の変化を部分空間上で表現し、様々な視点からの物体の認識を可能としている。また、認識の際、データベース中の莫大な数の部分空間全てと比較を行うと処理時間が膨大なものになるため、近似最近傍探索を用いて高速に類似度の高い部分空間を探索し、物体を認識する。

### 3.1 局所特徴量の選定

複数の局所特徴量から部分空間を作成する上で重要なことは、どのような局所特徴量の集合から作成するかということである。本手法では、撮影角度を連続して変化させながら物体を撮影し、複数のフレーム画像に渡ってある程度一貫して得られるような局所領域に注目する。なぜなら、そのような局所特徴量の集合から部分空間を作成することで、部分空間上に様々な撮影角度に対応する局所特徴量の変化を表現することができるからである。

図 1 に、そのような局所特徴量の集合を模式的に表す。黒い点が各フレーム画像における局所特徴量である。各局所特徴量の対応づけは、連続するフレーム画像において、 $i$  フレーム目の画像の各特徴量と  $i+1$  フレーム目の画像の全ての特徴量との距離計算をすることで行う。 $i$  フレーム目の特徴量を  $f$ 、 $i+1$  フレーム目の  $f$  と最も近い特徴量を  $g_1$ 、2 番目に近い特徴量を  $g_2$  とし、 $f$  と  $g$  のユークリッド距離を  $d(f, g)$  とすると、 $d(f, g_1) < d(f, g_2)/2$  を満たす場合にのみ対応付けを行う。これを連続する全てのフレーム画像間で行い、局所特徴量の軌跡を得る。以下、このような処理をトレース処理とよぶ。

### 3.2 部分空間の作成・登録処理

トレース処理によって得られた局所特徴量の集合から、KL 展開を用いて部分空間を作成する。このとき、トレース処理によって対応付けられる軌跡の長さは、図 1 の (a), (b), (c) のように様々である。軌跡の短い局所特徴量の集合からも部分空間を作成し、データベースに登録すると、その数は膨大なもの

となる．そこで，軌跡の長さの閾値  $T$  を設け，トレース処理が  $T$  フレーム以上続いたもののみ部分空間を作成し，データベースに登録する．作成した部分空間は，局所特微量の変化を近似的によく表現することができ，部分空間を作成する際には存在しないフレーム間の画像の局所特微量も，補間的に表現することができる．

KL 展開により局所特微量の集合から部分空間を作成する処理を次に示す．まず，トレース処理で対応づいた多数の局所特微量  $x$  から，それらの自己相関行列  $R$  を

$$R = E\{xx^T\}$$

によって計算する．そして， $\lambda_i (\lambda_i \geq \lambda_j \text{ if } i > j)$  を  $R$  の固有値， $u_i$  をそれに対する固有ベクトルとすると，

$$Ru_i = \lambda_i u_i$$

の固有値問題を解くことで固有ベクトルが求まる．求めたい部分空間の次元数を  $n$  とすると，上位  $n$  個の大きな固有値に対応する固有ベクトル

$$\{u_1, u_2, \dots, u_n\}$$

がその部分空間の基底となる．このとき，部分空間の次元数  $n$  が小さいほどメモリ容量は削減することができる．データベースには，この部分空間の基底ベクトルと物体 ID を登録する．

### 3.3 距離計算による認識処理

一般的に，局所特微量と部分空間の比較には，その射影長による類似度を用いる．類似度  $S$  は，局所特微量の特徴ベクトルを  $q$ ，射影する部分空間を成す  $n$  個の固有ベクトルを  $\{u_1, u_2, \dots, u_n\}$  とすると，

$$\|S\|^2 = \sum_{i=1}^n (q^T u_i)^2$$

より求めることができる．そして，最も類似度が高い部分空間を持つ物体 ID に投票処理を行う．これをクエリ画像から得られる全ての局所特微量について行い，最終的に得票数が最大となった物体 ID を認識結果として出力する．このとき，データベースに登録してある部分空間の数が物体ごとに異なるため，各得票数を物体ごとの部分空間の数で正規化する．物体  $j$  の部分空間の数を  $N_j$ ，得票数を  $v_j$  とすると，得票数を正規化した後のスコアは， $w_j = v_j / \sqrt{N_j}$  となる．

しかし，この一般的な類似度の求め方では，データベース中の部分空間の数が莫大なため，部分空間の次元が低次元であっても，その処理時間は膨大なものになってしまう．そこで本稿では，近似最近傍探索を用いて，高速に類似度の高い部分空間を探索する方法を述べる．

一般的な部分空間との類似度を，単に距離計算のみによって得ることは容易ではない．しかし，部分空間の次元が 1 次元の場合，クエリから得られた局所特微量と部分空間との類似度は，その基底との距離計算を行うことで求めることができる．各部分空間の基底の長さを正規化 ( $\|u_i\| = 1$ ) とすると，

$$\begin{aligned} \|q - u_1\|^2 &= \|q\|^2 - 2(q^T u_1) + \|u_1\|^2 \\ &\propto -q^T u_1 \\ &= -\|S\| \end{aligned}$$

が成り立つ．即ち，局所特微量と部分空間の基底との距離が近ければ近いほど，その部分空間との類似度は高くなるといえる．距離計算による最近傍点は，ANN などの近似最近傍探索の手法によって高速に求めることができる．

通常，部分空間の次元数を 1 次元にすることは，情報の欠損が大きすぎると考えられる．しかし，本手法のように投票処理によって物体の認識を行う場合は，それぞれの部分空間の識別精度がある程度低くても良い．なぜなら，識別精度が低くても，誤投票は分散するので，正解となる物体 ID の得票数のほうが他の物体 ID のそれより多くなると考えられるからである．本提案手法では，最も固有値の大きな固有ベクトルのみを基底とする部分空間を用いて，物体の認識を行う．

### 3.4 認識処理における 2 段階化

先に示した距離計算による探索方法は，1 次元の部分空間との類似度を比較する場合にしか用いることができない．しかし，部分空間の次元数を多く用いたほうが，部分空間一つ一つの識別性能が上がり，認識率の向上が見込まれるのも事実である．そこで，2 次元以上の部分空間を用いて，高速な認識を行うことを考える．

2 段階の処理を行うことで，2 次元以上の部分空間を用いて高速な認識を行う．処理の手順は単純である．1 段階目では，各部分空間の 1 次元のみを用いて近似最近傍探索を行い，類似度が高い  $k$  個の部分空間を求める．そして，2 段階目の処理では，絞り込んだ  $k$  個の  $n$  次元 ( $n > 1$ ) の部分空間に対して射影を行い，類似度を計算する．こうすることで，すべての部分空間に射影するよりも高速に，1 次元の部分空間のみを用いる場合よりも最適な部分空間を探索することができる．

### 3.5 相互部分空間法への拡張

相互部分空間法 (Mutual Subspace Method) は，クエリとして動画画像が与えられた場合に用いることができる．クエリ画像から局所特微量を抽出し，データベースを作成するときと同様にトレース処理を行って，局所特微量の対応付けを行う．そして， $T_q$  フレーム以上軌跡が続いた局所特微量の集合のみから部分空間を作成する．相互部分空間法では，部分空間同士の比較となるため，高い認識率を見込むことができる．

一般に相互部分空間法における類似度は，部分空間同士のなす正準角を算出することによって求める．しかし，その算出方法は単純ではなく，高速に行うことは困難である．そこで，相互部分空間法においても，近似最近傍探索を用いて高速な処理を行うことを考える．前述の手法と同様に，データベース・クエリとして扱う部分空間の次元数をともに 1 次元のみとする．このとき類似度は，その基底同士の距離を算出することで比較することができる．



図 2 データベースに登録した物体の一例

表 1 データベースに関するデータ

物体の数	1,002
画像数/1 物体	2,335
局所特徴量の数/1 物体	$1.14 \times 10^6$
トレース処理によって得た局所特徴量の数/1 物体	$3.0 \times 10^5$
部分空間の数/1 物体	1,567

## 4. 実験

### 4.1 実験準備

本実験では、ミニカーやフィギュア、ぬいぐるみ、日用雑貨など 1002 個の物体を用いて、提案手法の有効性を評価するための実験を行った。図 2 に、用いた物体のうちのいくつかを示す。これらの物体の中には、形状や模様が似ているものも多く、その識別は容易ではない。また、フィギュアなどには尖った部分が多数あり、少しの視点の変化によって物体の見え方が変化するものもある。

これら 1002 個の 3 次元物体をターンテーブルで 1 回転させ、正面、上 15 度、上 30 度からウェブカメラを用いて撮影し、フレーム画像（解像度：VGA）を取得した。そして、これらの画像から SIFT 特徴量 [6] を抽出し、トレース処理を行って部分空間を作成し、データベースとした (D1)。ここで、部分空間を作成する局所特徴量の集合は、トレース処理によって得られる軌跡の長さ  $T$  が  $T \geq 50$  を満たすもののみとした。表 1 に、データベースの詳細を示す。

次に本実験で用いたクエリについて説明する。クエリに用いる物体として、データベースに用いた 1002 物体のうち、無作為に 850 物体を選んだ。そして、より現実的なクエリを作成するために、手で持って物体を回転させて動画を撮影し、フレーム画像を取得した。図 3 に、これらのうちいくつかを示す。クエリとしては、フレーム画像単体を用いるもの (Q1) と、フレーム画像全体を用いるもの (Q2) の 2 種類を用意した。Q1 では、各物体のフレーム画像のうち 5 枚の画像を無作為に選択し、それらをクエリ画像として用いた。また Q2 では、各物体の全てのフレーム画像を用いてトレース処理を行い、その軌跡が  $T_q \in \{10, 20, 30, 40, 50\}$  フレーム以上のものから部分空間を作成し、クエリとした。表 2 に、クエリの詳細を示す。

本実験では、近似最近傍探索の手法として、ANN [11] を用い



図 3 クエリ用に撮影した物体の一例

表 2 クエリに関するデータ

	Q1	Q2
物体の数	850	850
画像数/1 物体	5	600
クエリの数	4250	850
局所特徴量の数/1 クエリ	350	$2.31 \times 10^5$
トレース処理によって得た局所特徴量の数/1 クエリ	—	$3.2 \times 10^3 (T_q = 50)$ $\sim 3.8 \times 10^4 (T_q = 10)$
部分空間の数/1 クエリ	—	$50 (T_q = 50)$ $\sim 1,950 (T_q = 10)$

た。ANN では、パラメータ  $\epsilon$  を用いて、探索する超球の半径を  $1/(1 + \epsilon)$  倍に縮小することで、近似の度合いを調節することができる。 $\epsilon$  の値が大きいほど、近似の度合いは強く、正しい最近傍が得られないというリスクは負うものの、大幅な処理時間の短縮が可能となる。

### 4.2 実験 1

まず、データベースを部分空間にするものの有効性を調べるための実験を行った。データベース D1 との比較のために、トレース処理によって得られた局所特徴量の集合から 4 フレームおきに登録したデータベース (D2) を用意した。D2 に登録された 1 物体あたりの局所特徴量数は、 $6.0 \times 10^4$  となった。それぞれのデータベースに対して認識処理を行うことで、部分空間を用いることの有効性を評価する。また、これらの処理には大変時間がかかるため、クエリ Q1 中の 850 物体の中から無作為に 100 物体を選択してクエリ (Q3) として実験を行った。

図 4 に、データベース D1 の部分空間の次元数を 1 次元から 10 次元まで変化させたときの、認識率と処理時間を示す。次元数が増えるにつれて認識率は増加傾向にあることが見てとれる。また、 $n = 9$  のとき、最も高い認識率が得られ、43.8% となった。また、データベース D2 を用いた場合の認識率は、37.0% であった。このことより、局所特徴量をそのままデータベースに登録するよりも、部分空間を作成してデータベースに登録するほうが認識率が向上することがわかった。

また、メモリ使用量を調べると、データベース D1 が約  $(750 \times n)$ MB ( $n$  は部分空間の次元数) であるのに対し、データベース D2 は約 15GB であった。例えば、部分空間の次元数が 2 次元の場合、メモリ使用量は約 1.5GB であり、データベース D2 の 10 分の 1 に削減することができる。このことより、メモ

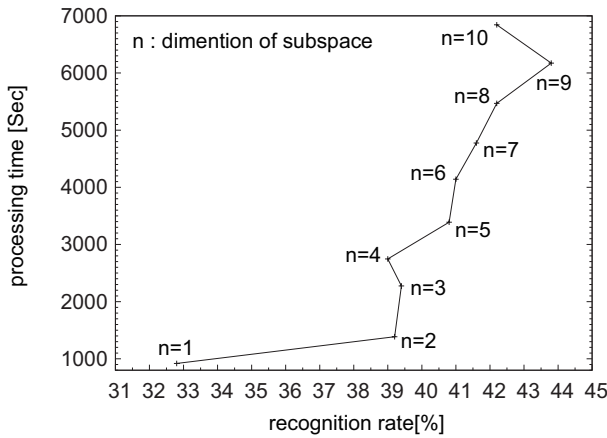


図 4 部分空間に射影したときの認識率と処理時間

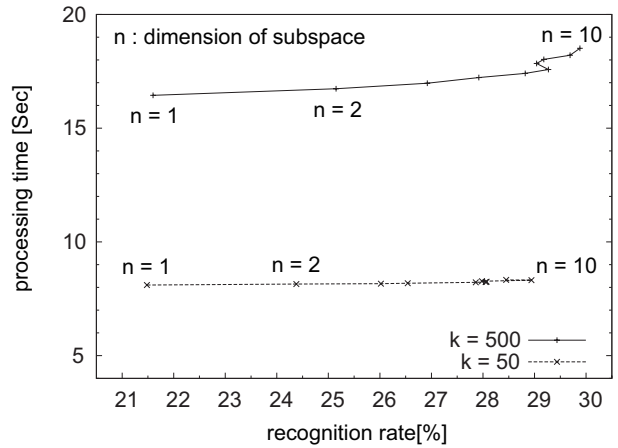


図 6 2段階処理による認識率と処理時間

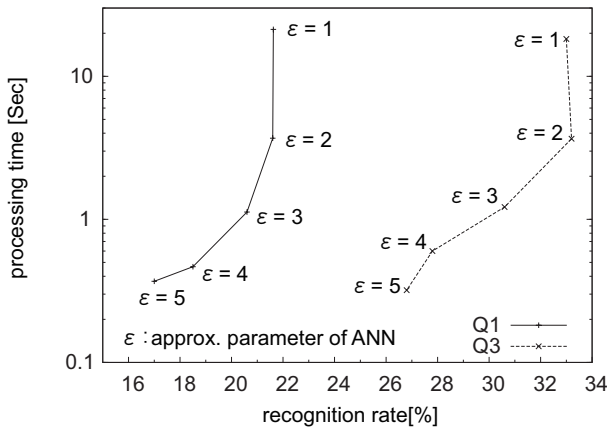


図 5 ANN を用いたときの認識率と処理時間

り使用量の面においても、部分空間を作成したほうが良いことがわかる。

#### 4.3 実験 2

次に、データベース D1 中の部分空間の次元数を 1 次元にして近似最近傍探索を用いて物体の認識を行うことで、処理時間や認識率にどのような影響があるのかを調べた。元クエリ Q1 とクエリ Q3 を用いて実験を行った。

図 5 に、ANN のパラメータ  $\epsilon \in \{1, 2, 3, 4, 5\}$  を変化させて物体の認識を行った場合のクエリ Q1, Q3 の認識率と処理時間を示す。クエリ Q1 と Q3 の認識率に大きな差が生じているが、これは Q1 から無作為に選択した Q3 が、偶然認識のしやすいものであったためである。

クエリ Q1, Q3 とともに、 $\epsilon = 2$  までは認識率を同程度に保ちながら、処理時間を短縮することができている。また、実験 1 の 1 次元の部分空間に射影した場合と Q3 の  $\epsilon = 2$  のときを比較すると、処理時間を 300 分の 1 に短縮することができた。

ここで、認識に成功したクエリと認識に失敗したクエリの傾向について考察する。認識に最も成功しやすいものは、缶などの円柱形のものであった。物体の形状に凹凸がなく、テクスチャも豊富なためであると考えられる。また、ミニカーにおいては、救急車やパトカー、特殊作業車など他の車との差異がはっきりしているものは認識率の良い結果となったが、乗用車などは物体間の差異があまりなく、認識率の低い結果となった。

フィギュアにおいては、正面からはっきり撮影されているもののほうが、その他の角度から撮影されているものより認識率が良かった。フィギュアの後方から撮影されたものは、大部分が髪の毛であったり着ている服であるため、特徴量が抽出されにくく、他のフィギュアとの区別が付きにくいと考えられる。その他に、クエリ画像中に撮影者の手が大きく映っているものは、認識に失敗する傾向にあった。

#### 4.4 実験 3

部分空間の 1 次元のみを用いて近似最近傍探索を行うことは、処理時間の短縮に有効であるが、認識率においては多次元の部分空間を用いた場合に及ばない。そこで、認識率の向上を目的として 2 段階処理の実験を行う。クエリには Q1 を用いた。

ANN のパラメータ  $\epsilon$  には、実験 2 で認識率が良く処理時間も短縮することができた  $\epsilon = 2$  を用いた。また、パラメータとして、1 段階目で ANN によって絞り込む部分空間の数  $k \in \{50, 500\}$  と、2 段階目で射影する部分空間の次元  $n \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  を変化させた。

図 6 に、認識率と処理時間の結果を示す。 $k = 50$  においては、 $n$  が大きくなって処理時間はほとんど変わらなかった。これは、絞り込む候補が少なく、2 段階目の処理にほとんど時間がかからないためである。2 段階目の処理にかかった時間は、 $n = 1$  のとき 0.028 秒で、 $n = 10$  のときは 0.251 秒であった。一方、 $k = 500$  においては、 $n$  の値が大きくなるにつれて、処理時間が増大している。これは、 $k$  が大きい場合には射影する部分空間の数が多くなり、その際に必要な処理時間が無視できなくなったためである。実際に  $k = 500$  のときの 2 段階目の処理時間は、 $n = 1$  のとき 0.387 秒、 $n = 10$  のとき 2.457 秒であった。

また、実験 2 の  $\epsilon = 2$  の結果と比較すると、処理時間は少しかかるが、認識率は向上することが分かった。更にパラメータを様々に変化させてクエリ Q3 を用いて実験を行った結果、 $k = 50, n = 5, \epsilon = 5$  のとき、0.6 秒ほどの処理時間で 33.4% の認識率を得ることができた。このことより、パラメータ次第では実験 2 の結果より高速で、高精度な処理が可能であることが分かった。

ここで、 $k = 500, n = 10, \epsilon = 2$  のときの、認識結果として



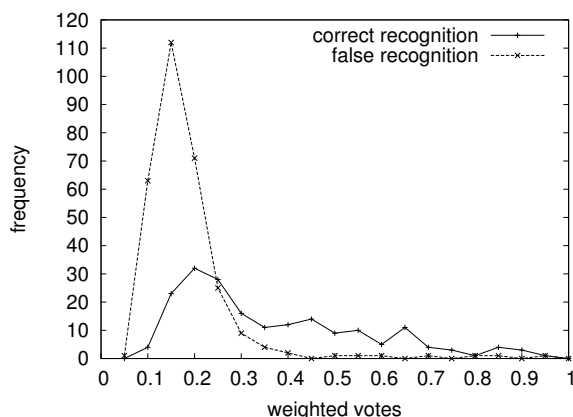


図7 認識結果別のスコアの頻度

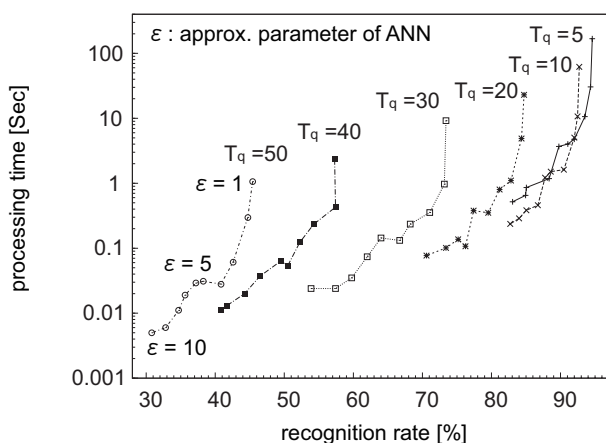


図8 相互部分空間法の認識率

出力されたものの得票数の傾向を調べる．図7に，データベース中の部分空間の数で正規化した得票数(スコア)ごとの頻度を示す．図7より誤認識の場合，認識結果として出力されたもののスコアは低い値に集中していることがわかる．これは，誤認識の場合，投票が分散し，ある特定の物体に得票数が集まらないためである．また，正しく認識された場合は，スコアが低いものもあるものの，誤認識の場合と比べればスコアが高いものが多い．このことより，正しく認識するには，ある程度のスコアが必要であると考えられる．

#### 4.5 実験4

最後に，相互部分空間法においても近似最近傍探索を用いて実験を行った．クエリはQ2を用いた．図8に，ANNの $\epsilon$ を変化させた時の認識率と処理時間を示す． $T_q = 5$ ， $\epsilon = 1$ のときに94.6%と最も高い認識率となり，そのときの処理時間は168.2秒であった．更に $\epsilon = 4$ のとき，認識率が92.0%と少し低下したが，4.7秒の処理時間で認識を行うことができた．また， $T_q = 10$ ， $\epsilon = 4$ のとき，1.6秒で90%以上の認識率を得ることができた．

他の実験の結果と比較すると，認識率は最も良いときで60%以上向上し，処理時間も短く保たれている．また，実験3の最も良かった29.9%と同程度の認識率を，5ミリ秒の処理時間で得ることができている．これらの結果より，相互部分空間法で他のどの手法よりも，認識率・処理時間の両面で最も良いということがいえる．



図9 誤認識となった物体の一例

最後に， $T_q = 5$ ， $\epsilon = 1$ で誤認識となったものについて述べる．図9に誤認識の例を示す．傾向としては，ミニカーの乗用車など酷似している物体があるものや，果物などテクスチャがあまりないもの，白っぽいものなどが挙げられる．

## 5. まとめ・今後の課題

本研究では，大規模な部分空間を用いて3次元物体の認識を高速に行う手法を提案した．実験結果より，1次元の部分空間を用いて近似最近傍探索を行うことで，認識率をほとんど低下させることなく，処理時間を大幅に短縮することができた．特に相互部分空間法においては，1秒程度の短い処理時間で90%以上の認識を行うことができた．

今後の課題としては，カーネル相互部分空間法などの高度な部分空間法に提案手法を適用することで，更なる認識率の向上と，処理時間の高速化を目指す．

謝辞 本研究の一部は科学研究費補助金基盤研究(B)(22300062)の補助による．

## 文献

- [1] D. G. Lowe: "Local feature view clustering for 3D object recognition", CVPR2001 (2001).
- [2] K. Inoue, K. Kise: "Compressed Representation of Feature Vectors Using a Bloomier Filter and Its Application to Specific Object Recognition", Proc. of WS-LAVD2009 (2009).
- [3] 村瀬: "パラメトリック固有空間法による画像認識", 電子情報通信学会誌, **85**, 4, pp. 252-258 (2002).
- [4] 前田, 山口: "相互部分空間法による三次元パターンマッチングにおける正準角の性質", 情報処理学会研究報告, 2007-CVIM-160 (2007).
- [5] 福井, 山口: "部分空間法の理論拡張と物体認識への応用", 情報処理学会論文誌, **46**, 15, pp. 21-34 (2005).
- [6] D. Lowe: "Distinctive image features from scaleinvariant keypoints", International Journal of Computer Vision, **60**, pp. 91-110 (2004).
- [7] 柏木, 外山, 黄瀬: "局所特徴量の部分空間を用いた3次元物体認識におけるメモリ削減手法", 電子情報通信学会技術研究報告, PRMU2010-19, **110**, 27, pp. 97-102 (2010).
- [8] S. Kim, I. S. Kweon: "Scalable representation for 3D object recognition using feature sharing and view clustering", Pattern Recognition, **41**, pp. 754-773 (2008).
- [9] S. Watanabe, N. Pakvasa: "Sunspace method in pattern recognition", Proc. 1st IJCP (1973).
- [10] 福井, 山口: "カーネル非線形制約相互部分空間法による物体認識", 電子情報通信学会論文誌, **88**, 8, pp. 1349-1356 (2005).
- [11] R. S. S. Arya, D. M. Mount and A. Y. Wu: "An optimal algorithm for approximate nearest neighbor searching", Journal of the ACM, **45**, 6, pp. 891-923 (1998).
- [12] A. Andoni, P. Indyk: "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions", Comm. of the ACM, **51**, 1, pp. 117-122 (2008).
- [13] R. Basri, T. Hassner, L. Zelnik-Manor: "Approximate nearest subspace search", IEEE Trans. PAMI, **33**, 2, pp. 266-278 (2011).