

# 特徴空間の次元数と部分空間数にスケーラブルな 近似最近傍部分空間探索

小西 将貴<sup>†</sup> 岩村 雅一<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> konishi@m.cs.osakafu-u.ac.jp, {masa,kise}@cs.osakafu-u.ac.jp

あらまし パターンを部分空間で表現する方法がコンピュータビジョンやパターン認識の分野で広く利用されている。扱う問題の大規模化に伴い、最も近い部分空間を探索する最近傍部分空間探索のスケーラビリティの向上が求められている。その障害になるのは、部分空間同士の距離はグラスマン多様体上で定義される距離で測られるため、ユークリッド空間上の距離でよく用いられる近似最近傍探索手法が直接適用できないことである。近似を導入した部分空間の効率的な探索を目指す手法は提案されているが、大規模なデータセットや高次元のデータに有効ではない。本稿では、グラスマン多様体上の距離の計算を複数のユークリッド距離の計算に分解することで、部分空間同士の類似度の効率的な近似計算を実現する。実験の結果、部分空間の探索精度を 3% 落とすことで従来手法と比べて、約 800 倍の高速化が実現できた。

キーワード 部分空間, 近似最近傍探索, グラスマン多様体, グラスマンカーネル, 正準角

## 1. ま え が き

コンピュータビジョンやパターン認識の分野では、ユークリッド空間上の線形部分空間で何かを表現するという手法が注目されており、行動認識、顔認識、歩行者の検出など幅広く用いられている [1]~[4]。パターンを部分空間で表して、パターン認識を行う場合、クエリの部分空間に最も近い部分空間をデータベース中から見つける問題に帰着する。この問題は最近傍部分空間探索問題 (Near Subspace Search; NSS) と呼ばれる。近年のパターン認識における学習データの大規模化を考慮すると、高次元かつ大規模なデータベースに対して有効な探索手法が必要となる。そこで NSS の類似計算に近似を導入した近似最近傍部分空間探索 (Approximate Nearest Subspace Search; ANSS) の効率化を目指す。

ANSS 問題の高速化を阻む主因は、部分空間同士の距離がユークリッド距離のような一般的に使われている距離で測れないことである。部分空間同士の距離は、部分空間の集合であるグラスマン多様体上で定義される距離を用いることで測ることができる。グラスマン多様体上での距離は様々なものが定義されている。本稿では、グラスマン多様体上で定義される距離の総称をグラスマン距離と呼ぶ。クエリ点に最も近い点を探索する近似最近傍探索はユークリッド距離のような一般の距離を前提にしているため、この研究成果をそのまま ANSS には適用できず、新たな手法が求められる。

このような問題に対処するために、ANSS には 2 種類のアプローチが提案されている。ひとつは、グラスマン多様体上で近似最近傍探索手法を用いる方法である。この方法はグラスマン

距離の一種である測地距離を用いるため、計算コストの高い主成分分析 (PCA) や特異値分解 (SVD) を必要とする。そのため、どんなに優れた近似最近傍探索手法を用いてもこのアプローチで効率的な ANSS を実現できない。もうひとつのアプローチはグラスマン多様体からユークリッド空間へ埋め込むものである [5]~[7]。すなわち、グラスマン多様体上の点同士の問題から、高次元のユークリッド空間上の点同士の問題へと変換し、近似最近傍探索を利用して効率的な ANSS を目指す。しかし、変換されたユークリッド空間が非常に高次元であるため、既存のどの近似最近傍探索手法も効率的に使用できない。

本稿では、特徴空間の次元数と部分空間数の増加に対する処理時間の増加を抑えた ANSS 手法を提案する。ある種のグラスマン距離は複数のユークリッド距離の計算で表現できることに着目し、近似最近傍探索でグラスマン距離を高速に計算する。

## 2. 準 備

本節では、提案手法を理解するために必要なグラスマン多様体などの事項について説明する。

### 2.1 単位ベクトルの距離と内積の関係

グラスマン距離の近似計算に単位ベクトルの距離と内積の関係が重要となるので、まずそれについて述べる。2 つの単位ベクトル  $\mathbf{a}$ ,  $\mathbf{b}$  のユークリッド距離の 2 乗は次のようになる。

$$d_{Euc}^2(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\mathbf{a}^\top \mathbf{b} \quad (1)$$

$\|\mathbf{a}\|^2 = \|\mathbf{b}\|^2 = 1$  であるので、

$$d_{Euc}^2(\mathbf{a}, \mathbf{b}) = 2 - 2\mathbf{a}^\top \mathbf{b} \quad (2)$$

となる。この場合、内積と距離は1対1の写像になっており、内積が大きくなれば、距離は小さくなるという関係にある。

## 2.2 正準角

最も近い部分空間を探索するために、部分空間間の距離を求める必要があるが、そのうちのいくつかは正準角を使って計算される。正準角の定義は以下の通りである。

[定義1]  $\mathbf{Y}_1, \mathbf{Y}_2$  を  $D \times m$  の直交基底行列とし、 $\mathbf{Y}_1, \mathbf{Y}_2$  の張る部分空間をそれぞれ  $\text{span}(\mathbf{Y}_1)$  と  $\text{span}(\mathbf{Y}_2)$  と表す。部分空間  $\text{span}(\mathbf{Y}_1)$  と  $\text{span}(\mathbf{Y}_2)$  のなす正準角  $0 \leq \theta_1 \leq \dots \leq \theta_m$  は  $\mathbf{Y}_1^\top \mathbf{Y}_2$  を以下のようにSVDによって得られる。

$$\mathbf{Y}_1^\top \mathbf{Y}_2 = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top \quad (3)$$

$\mathbf{\Sigma}$  は  $m \times m$  の対角行列で、対角成分は  $\mathbf{\Sigma} = \text{diag}(\cos \theta_1, \dots, \cos \theta_m)$  となっている。 $\mathbf{U}, \mathbf{V}$  はそれぞれ  $m \times m$  の直交基底行列である。

## 2.3 グラスマン多様体

[定義2] ユークリッド空間  $R^D$  上の  $m$  次元の線型部分空間の集合をグラスマン多様体  $G(m, D)$  と呼ぶ。

グラスマン多様体上では部分空間は点とみなすことができる。

## 2.4 グラスマン距離

グラスマン多様体上で定義されるグラスマン距離のうち自然な距離尺度として、測地距離 (geodesic distance) がある。これはグラスマン多様体上の点と点を結ぶ測地線の長さであり、正準角を使って、測地距離  $d_G$  は以下の様に表される。

$$d_G(\mathbf{Y}_1, \mathbf{Y}_2) = \left( \sum_i \theta_i^2 \right)^{1/2} \quad (4)$$

$\theta_i$  は第  $i$  番目の正準角である。測地距離は正準角を得るためにPCAまたはSVDを必要とするので、計算コストが大きい。

文献 [8], [9] では正準角を使った距離距離尺度がいくつか定義されている。その1つである projection metric を紹介する。

$$d_P(\mathbf{Y}_1, \mathbf{Y}_2) = \left( \sum_{i=1}^m \sin^2 \theta_i \right)^{1/2} = \left( m - \sum_{i=1}^m \cos^2 \theta_i \right)^{1/2} \quad (5)$$

文献 [8] で、projection metric がカーネル化されている。カーネル化された projection metric は projection kernel と呼ばれ、以下の式で計算される。

$$k_P(\mathbf{Y}_1, \mathbf{Y}_2) = \|\mathbf{Y}_1^\top \mathbf{Y}_2\|_F^2 \quad (6)$$

計算に正準角を必要としないので、projection metric は測地距離より計算コストが小さい。projection kernel は提案手法に望ましい性質を持っている。その性質については4節で説明する。

もうひとつ Grassmannian radial basis function (GRBF) kernel というグラスマン多様体上のカーネル関数が文献 [10] で提案されている。RBF kernel は次式で表される。

$$k_{RBF}(\mathbf{Y}_1, \mathbf{Y}_2) = \exp\left(\beta \|\mathbf{Y}_1^\top \mathbf{Y}_2\|_F^2\right), \beta > 0 \quad (7)$$

Grassmannian RBF kernel も同様に提案手法にとって望ましい性質を持っている。

## 3. 関連研究

ANSS の手法の1つに Basri らの手法 [5]~[7] がある。これ

以降は [11] に倣い、Basri ら手法を BHZ と呼ぶ。BHZ は部分空間を高次元空間上の点に写像し、空間同士の問題から点同士の問題へ変換し、近似最近傍探索の利用を実現している。以下にその手法について簡単に説明する。部分空間から点への写像は非常に高次元な空間への写像となる。元の特徴空間を  $d$  次元とすると、写像先の空間の次元数は  $d(d+1)/2$  次元となる。特徴空間が 256 次元、1024 次元の場合、写像先の空間はそれぞれ 32,896 次元、524,800 次元となる。これほどの高次元空間では近似最近傍探索はうまく機能せず、その処理時間は全探索する場合と同程度以上になる。

Wang らは ANSS の手法として Grassmannian-based Locality Hashing (GLH) [11] を提案している。GLH は近似最近傍探索手法である Locality Sensitive Hashing (LSH) をグラスマン多様体上で測地距離に用いている。正確には、ランダムなベクトルごとに、ベクトルと部分空間のなす角が閾値 ( $= \pi/6$ ) 以下になるかで部分空間を2つに分けている。しかし、GLH を大規模な ANSS に使うのは現実的ではない。それは、LSH が低速な近似最近傍探索手法であることと、高次元空間ではベクトル同士はほぼ直交しているので、ベクトル同士のなす角が閾値  $\pi/6$  以下になることはほとんどないことによる。さらに、測地距離を計算するたびに、計算コストがかかる PCA または SVD が必要という問題もある。

## 4. 提案手法

### 4.1 問題定義

まず、本稿で扱う最近傍部分空間探索問題は、データベース中に登録されている部分空間からクエリとして与えられた部分空間に最も近いものを見つけるものである。データベース中の部分空間とクエリの部分空間をそれぞれ  $\text{span}(\mathbf{P}_i)$ ,  $\text{span}(\mathbf{Q})$  とする。ここで、 $\mathbf{P}_i, \mathbf{Q}$  は  $D \times m$  の正規直交基底である。問題を定式化すると

$$i^* = \arg \min_i \text{dist}(\mathbf{P}_i, \mathbf{Q}) \quad (8)$$

または

$$i^* = \arg \max_i \text{sim}(\mathbf{P}_i, \mathbf{Q}) \quad (9)$$

となる。 $i^*$  はクエリの部分空間に最も近い部分空間のIDである。また  $\text{dist}(\cdot, \cdot)$ ,  $\text{sim}(\cdot, \cdot)$  はそれぞれ部分空間同士の距離関数、類似度関数である。本稿では、式 (6) の projection kernel  $k_P(\cdot, \cdot)$  と式 (7) の Grassmannian RBF kernel  $k_{RBF}(\cdot, \cdot)$  を式 (9) の類似度関数として使用する。

### 4.2 近似距離計算

提案手法は部分空間同士の距離を複数のユークリッド距離の計算で求める。類似度関数である  $k_P(\cdot, \cdot)$  と  $k_{RBF}(\cdot, \cdot)$  を分解し、それが可能であることを示す。まず、 $k_P(\cdot, \cdot)$  について説明する。 $\mathbf{P}_i = [\mathbf{p}_{i1}, \dots, \mathbf{p}_{im}]$ ,  $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_m]$  とすると、 $\mathbf{P}_i^\top \mathbf{Q}$  の要素は

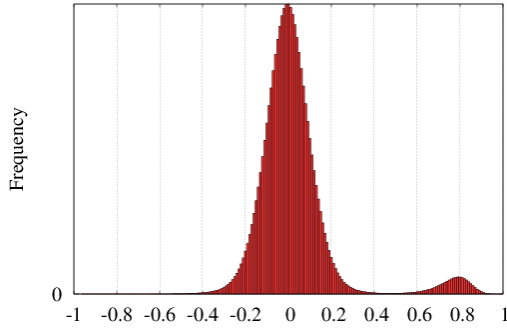


図 1: 内積のヒストグラムの一例

$$\mathbf{P}_i^\top \mathbf{Q} = \begin{pmatrix} \mathbf{p}_{i1}^\top \mathbf{q}_1 & \mathbf{p}_{i1}^\top \mathbf{q}_2 & \cdots & \mathbf{p}_{i1}^\top \mathbf{q}_m \\ \mathbf{p}_{i2}^\top \mathbf{q}_1 & \mathbf{p}_{i2}^\top \mathbf{q}_2 & \cdots & \mathbf{p}_{i2}^\top \mathbf{q}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{p}_{im_1}^\top \mathbf{q}_1 & \mathbf{p}_{im_1}^\top \mathbf{q}_2 & \cdots & \mathbf{p}_{im_1}^\top \mathbf{q}_m \end{pmatrix} \quad (10)$$

となる。よって、式 (6) は次のように表せる。

$$k_P(\mathbf{P}_i, \mathbf{Q}) = \|\mathbf{P}_i^\top \mathbf{Q}\|_F^2 = \sum_{s=1}^m \sum_{t=1}^m (\mathbf{p}_{is}^\top \mathbf{q}_t)^2 \quad (11)$$

式 (11) の右辺は  $m^2$  個の内積計算の和であるが、すべての内積の値が式 (11) の値に等しく影響を及ぼしているわけではない。図??に示すように、ほとんどの内積は 0 付近に分布しており、一部の内積のみが比較的大きな絶対値をとる。それは、固有ベクトル  $\mathbf{P}_i = [\mathbf{p}_{i1} \dots \mathbf{p}_{im}]$  が正規直交基底であり、ノルムが 1 のクエリベクトルを正規直交基底  $P_i$  に射影したときの射影長は 1 になるからである。ここで式 (9) を使用する目的が、ある部分空間に最も近い部分空間を見つけることであることを考慮すれば、部分空間間の距離の大小関係が入れ替わらない程度に式 (11) を近似しても問題は起らない。つまり、式 (11) の 0 に近い内積を無視して、絶対値の大きい内積のみを計算することで近似的に projection kernel の値を計算できる。

この近似計算を実現するために、2.1 節で述べた、「ベクトルの内積の大きいものを見つけることと、ベクトルの距離が近いものを見つけることは等価」であることを利用する。この原理を用いれば、近似最近傍探索手法を用いて内積の大きいベクトルを見つけることができる。

ただし、類似度の近似計算に必要なのは絶対値の大きい内積であることを注意しなくてはならない。つまり、距離が小さいベクトルだけではなく、距離が大きいベクトルも式 (11) では支配的である。この問題を解決するために、クエリベクトル  $\mathbf{q}$  に加えて  $-\mathbf{q}$  もクエリベクトルとして使う。  $\mathbf{q}$  をクエリとすることで、近似最近傍探索で  $\mathbf{q}$  に近いベクトルを見つけることができるのは明らかである。  $-\mathbf{q}$  は単位超球上で  $\mathbf{q}$  から最も遠いベクトルなので、  $-\mathbf{q}$  をクエリとして近似最近傍探索をすることで  $\mathbf{q}$  から最も遠いベクトルから探索することができる。

このように符号の違う 2 つのクエリベクトルを用いることで、  $\mathbf{P}_i^\top \mathbf{Q}$  の要素 (内積) のうち、絶対値の大きいものを近似最近傍探索で上位  $K$  個を取り出し、対応する部分空間の ID に内積の 2 乗を投票する。そして、最終的に最も得票の多い部分空間

を最近傍と判断する。  $\mathbf{Q}$  の基底が  $m$  本の場合、1 本の基底ベクトルあたり上位  $k$  個を選ぶ。更に符号が逆のベクトルも含めると、基底ベクトルあたり  $2k$  個のベクトルを近似最近傍探索で求める。したがって、全体で  $K = 2km$  個のベクトルを検索することになる。データベース中の部分空間数が増加しても、  $K$  を加減することで探索にかかる時間をコントロールできる。このような理由から、提案手法は部分空間数にスケラブルな探索が可能である。

また、前述のように、  $\mathbf{p}_{is}^\top \mathbf{q}_t$  の多くは 0 付近の値を取る。そのため、特徴空間が高次元であっても、類似度関数として式 (11) を計算する際にほとんどの内積は 0 に近いとして無視できる。そのため、提案手法は特徴空間次元数にスケラブルになる。

また、Grassmannian RBF kernel  $k_{RBF}(\mathbf{P}_i, \mathbf{Q})$  も  $\mathbf{P}_i^\top \mathbf{Q}$  の要素の和で表せるので、  $k_P(\mathbf{P}_i, \mathbf{Q})$  と同様に近似計算ができる。

## 5. 実験・考察

提案手法の有効性を調べるために、物体認識実験と手書き文字認識実験を行い、従来手法と比較をした。実験で用いた各手法のパラメータを表 1 にまとめた。 projection kernel と Grassmannian RBF kernel に 4.2 節で説明した近似計算を導入した提案手法をそれぞれ approximate projection kernel (APK), approximate Grassmannian RBF kernel (ARBF) と呼ぶことにする。以下では projection kernel, Grassmannian RBF kernel, 測地距離をそれぞれ PK, RBF, GD と呼ぶこととする。全ての手法は C++ で実装した。また、実験に使用した計算機の CPU は Intel Xeon E54627 v2, 3.3GHz, メモリは 512GB である。提案手法や従来手法で近似最近傍探索が必要なものには Bucket Distance Hashing (BDH) [12] を用いた。 BHZ に BDH を適応した場合、探索空間の次元数が高すぎたためにうまく動作しなかった。そのため、全探索を代わりに用いた。 GLH はランダムベクトルと部分空間のなす角が閾値 ( $\pi/6$ ) 以下になるかどうかで、部分空間を 2 つにインデクシングしている。 GD, PK, RBF では全探索を使用した。

### 5.1 物体認識

物体認識実験では、41 方向の視点から撮影した 8 クラス 80 物体からなる公開データセット ETH-80 Image Set [13] を使用した。各クラスにはサブクラスが 10 個存在する。各物体について 41 視点から撮られた画像がある。画像サイズは  $256 \times 256$  である。マスク画像を用いて背景を除去し、グレースケールにした画像を用いる。これを最近傍法で  $16 \times 16$  に縮小し、ラスタ走査をして、256 次元の特徴ベクトルとする。各物体に含まれる 41 枚の画像のうち、奇数番目の画像 21 枚の画像を学習データとする。また、偶数番目の画像のうち、連続する 10 枚の画像を一つのテストデータとし、先頭となる画像を 1 つずつずらしていき、1 物体あたり 11 種類のテストデータを作成する。

使用した部分空間の次元数  $m$  は予備実験で決定した。類似度関数として PK を用いて、次元数  $m$  を変化させ、ETH-80 で認識実験をし、最も認識率が高くなる  $m$  を調べたところ、  $m = 7$  であった。他の手法でも部分空間の次元数は  $m = 7$  として実験した。データベース中の部分空間数は 80 個あるので、探索

表 1: 実験使用した手法の名前とパラメータ (\*が付いているのが提案手法)

略称	名前	使用パラメータ
GD	Geodesic distance	N/A
PK	Projection kernel	N/A
RBF	Grassmannian RBF kernel	$\beta = 1$
BHZ	Basri らの手法 [7]	N/A
GLH	Grassmanian-based locality hashing [11]	$S = 100, 500, 1000, 5000, 10000$ と $K = 1, 2, 3, 4, 5$ の組み合わせ
APK*	Approximate projection kernel	$k = 1, \dots, 2000$
ARDF*	Approximate Grassmannian RBF kernel	$k = 1, \dots, 2000$

表 2: 手書き文字認識実験における従来手法と比較したときの提案手法 (APK) の性能 (256 次元特徴量を使用した場合)

従来 手法	従来手法の結果		APK の結果			APK の従来手法との差	
	認識 [%]	処理時間 [ms]	認識率 [%]	処理時間 [ms]	$k$	認識率 [%]	処理時間の比 [倍]
GD	96.18	207.53	96.38	12.13	101	0.20	17.11
PK	99.31	152.61	98.22	29.11	301	-1.09	5.24
GLH	96.18	197.48	96.38	12.13	101	0.20	16.28
BHZ	99.31	9402.71	99.21	144.57	1201	-0.10	65.04

表 3: 手書き文字認識実験における従来手法と比較したときの提案手法 (APK) の性能 (1024 次元特徴量を使用した場合)

従来 手法	従来手法の結果		APK の結果			APK の従来手法との差	
	認識 [%]	処理時間 [ms]	認識率 [%]	処理時間 [ms]	$k$	認識率 [%]	処理時間の比 [倍]
GD	96.34	1017.2	96.31	26.22	101	-0.03	38.79
			93.28	1.26	1	-3.04	807.30
PK	99.31	1350.77	98.72	61	301	-0.59	22.14
GLH	96.34	191.81	96.31	26.22	101	-0.02	7.32
			93.28	1.26	1	-3.04	800.57
BHZ	99.31	747324	98.72	61	301	-0.59	12251.21

する内積の総数は 3920 個である。つまり、 $k = 280$  のときに提案手法の認識率は近似しない場合と等しくなる。

認識結果を図 2 に示す。図 2(a), 2(b) では、パラメータ  $k$  が増加すると提案手法である APK, ARBF の認識率、処理時間ともに単調増加している。APK, ARBF はそれぞれ  $k = 45$  で PK, RBF と同じ認識率となっている。しかし、近似計算の処理のオーバーヘッドにより APK, RBF の方が PK, RBF より処理時間がかかっている。認識率と処理時間の関係図である図 2(c) を見ると、提案手法が PK, RBF を除く従来手法を上回っていることがわかる。

## 5.2 文字認識

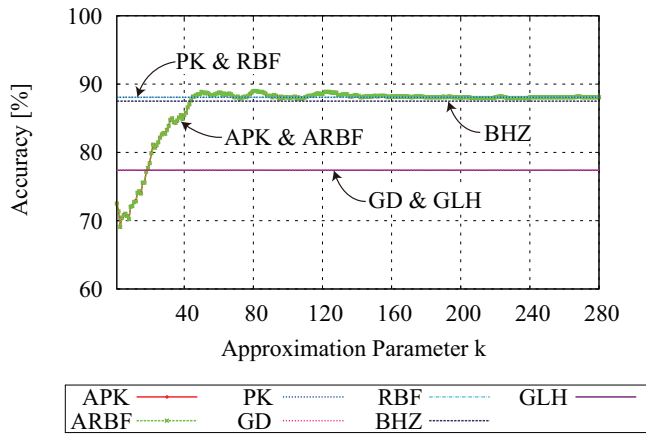
データセットとして手書き文字データベース ETL9B [14] を使用した。ETL9B は 3036 字種が含まれており、1 字種あたり 200 サンプルが用意されている。前半の 100 サンプルを学習データ、残りをテストデータとした。文字画像は  $64 \times 63$  画素で、2 値画像である。文献 [15] の手法を使い、 $64 \times 64$  画素に非線形正規化をした。特徴量は 2 つの特徴量を使用した。 $64 \times 64$  の画像の重複しない  $4 \times 4$  の 16 画素の輝度値の和を 1 次元とした 256 次元特徴量と、もうひとつは  $64 \times 64$  画素の画像の重複しない  $2 \times 2$  の 4 画素の輝度値の和を 1 次元とした 1024 次元特徴量である。

部分空間の次元数  $m$  は 5.1 節と同じ方法で決めようとした

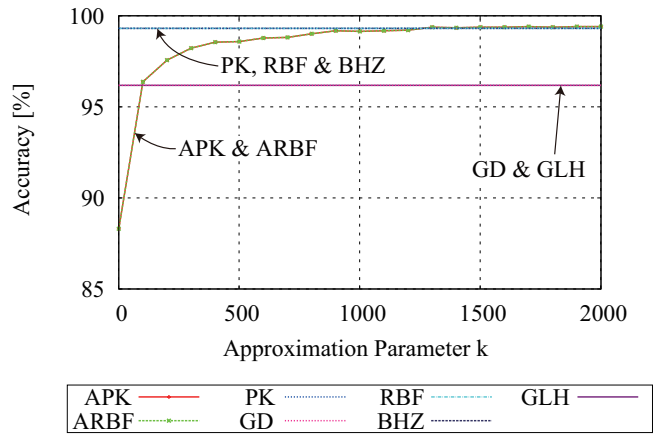
が、256 次元、1024 次元特徴量を使ったどちらの場合も  $m = 1$  のときに認識率が最も高くなった。本実験の目的は部分空間を利用した提案手法の有効性を確認することである。 $m = 1$  にするとベクトル同士の比較になってしまうため、部分空間の次元数は  $m = 5$  とした。従来法、提案手法ともに  $m = 5$  で 256 次元特徴量、1024 次元特徴量を使って実験をした。データベース中の部分空間数は 3036 個なので、探索する内積の総数は 75900 個である。提案手法は  $k = 7590$  のとき、近似のない場合と同じ結果になる。

256 次元、1024 次元特徴量を使った場合の認識結果をそれぞれ図 3, 4 に示す。256 次元特徴量の結果の図 3(a), 3(b) でも物体認識の結果と同様に、パラメータ  $k$  が増加すると提案手法である APK, ARBF の認識率、処理時間ともに単調増加している。 $k = 1301$  で APK, ARBF はともに PK, RBF と同じ認識率となっているが、物体認識と同様に、近似計算の処理のために処理時間がかかっている。図 3(c) の認識率と処理時間の関係をみると、提案手法が他の従来法よりも性能が優れていることがわかる。1024 次元特徴量の結果も同様の傾向であった。

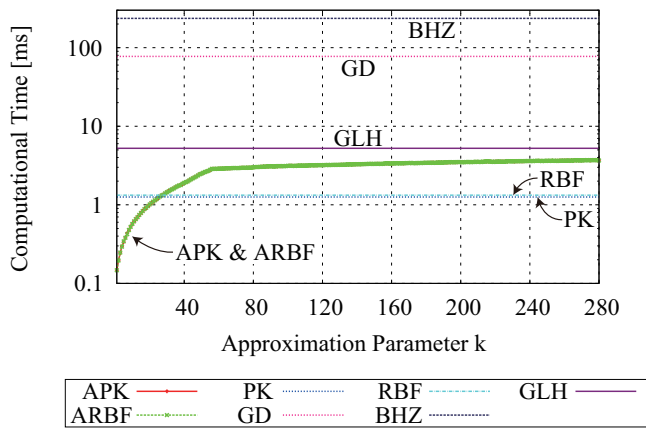
表 2, 3 に APK と従来手法が同じ程度の認識における処理時間の比をまとめた。256 次元の特徴量を使った場合、APK は GD と同程度の認識率のとき、処理速度は GD の 17.11 倍であった。同様に GLH, GLHM と同程度の認識率のときの処理



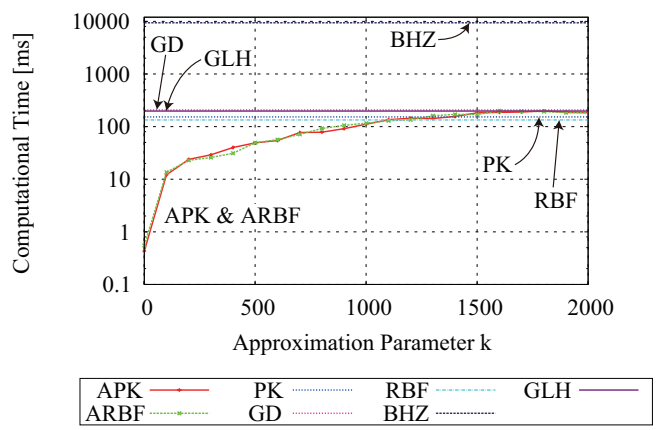
(a) k vs Accuracy



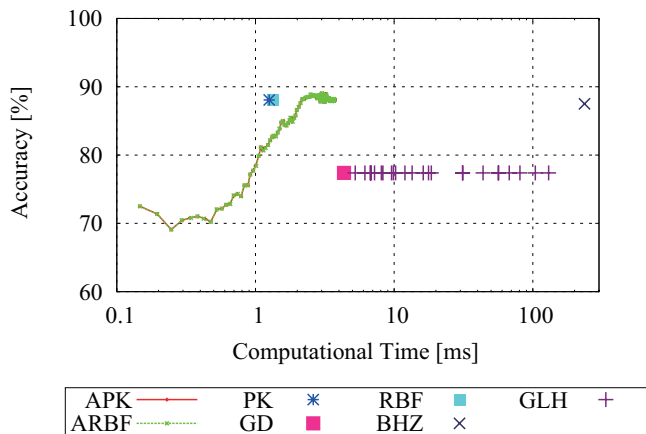
(a) k vs Accuracy



(b) k vs Time

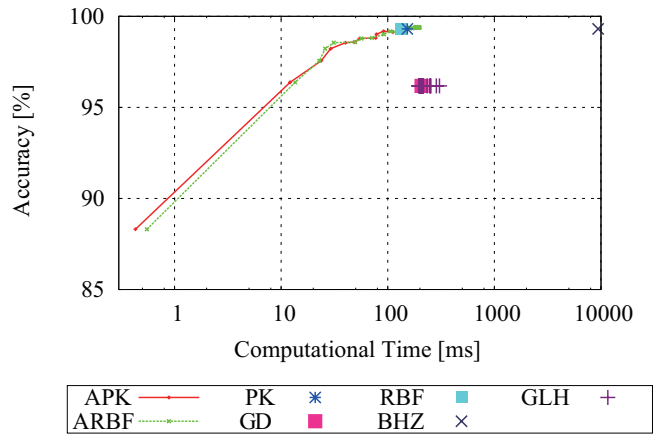


(b) k vs Time



(c) Time vs Accuracy

図 2: ETH-80 の実験結果



(c) Time vs Accuracy

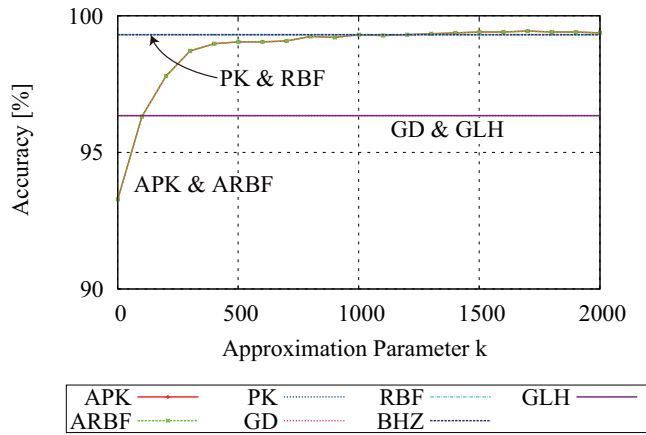
図 3: 256 次元特徴量を使った ETL9B の実験結果

時間を比べた場合、それぞれ処理速度は 16.28 倍, 15.46 倍となった。1024 次元の場合, APK は GD よりも認識率が 3.04% 低くすると, 処理速度は 807.30 倍速くなった。また GLH と比べた場合, 3.04% 認識率を落とすと, 処理速度は 800.57 倍になった。GLH 以外と比較した場合には, 特徴空間の次元が大きいほど提案手法との処理時間の差が大きくなった。

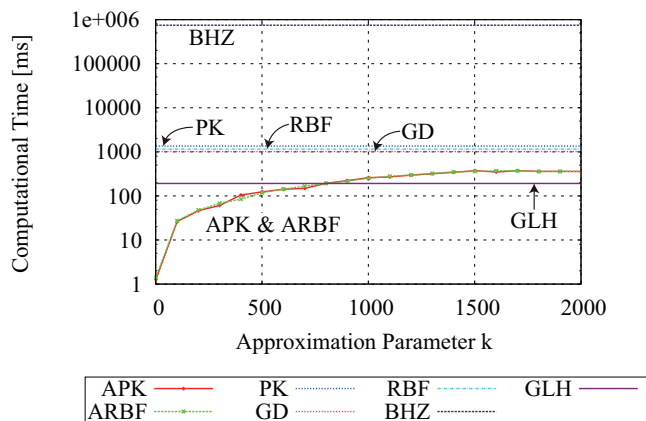
## 6. まとめ・今後の課題

本稿では特徴空間の次元数と部分空間数にスケーラブルな部

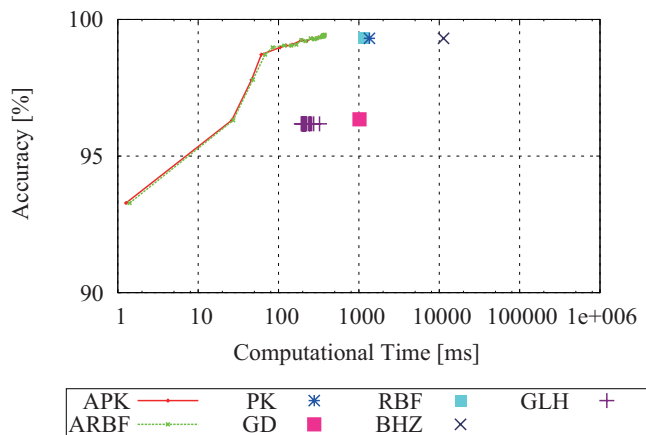
分空間の探索手法を提案した。グラスマン距離の計算を複数のユークリッド距離の計算に置き換え, 近似最近傍探索によって近似的に計算することで, 高速な部分空間の探索を実現した。文字認識実験では提案手法は従来手法と比べたとき, 認識率を 3% 程度落とすことで, 測地距離を比べて, 処理速度が 800 倍向上した。さらに特徴空間が高次元であるほど, 提案手法が有効であることが示された。今後は, さらに別のデータセットで実験を行うことで提案手法の有効性を調べることを予定している。



(a)  $k$  vs Accuracy



(b)  $k$  vs Time



(c) Time vs Accuracy

図 4: 1024 次元特徴量を使った ETL9B の実験結果

謝辞 本研究の一部は、JST CREST および日本学術振興会科学研究費補助金基盤研究 (A)(25240028) の補助による。

#### 文 献

- [1] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, "Statistical computations on grassmann and stiefel manifolds for image and video-based recognition," IEEE TPAMI, vol.33, no.11, pp.2273–2286, Nov. 2011.
- [2] A. Sanin, C. Sanderson, M.T. Harandi, and B.C. Lovell, "Spatio-temporal covariance descriptors for action and gesture recognition," Proc. WACV, pp.103–110, Jan. 2013.
- [3] L. Wang, X. Wang, and J. Feng, "Subspace distance analy-

sis with application to adaptive bayesian algorithm for face recognition," Pattern Recognition, vol.39, no.3, pp.456–464, March 2006.

- [4] M.T. Harandi, C. Sanderson, S. Shirazi, and B.C. Lovell, "Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching," Proc. CVPR, pp.2705–2712, June 2011.
- [5] R. Basri, T. Hassner, and L. Zelnik-Manor, "Approximate nearest subspace search with applications to pattern recognition," Proc. CVPR, pp.1–8, June 2007.
- [6] R. Basri, T. Hassner, and L. Zelnik-Manor, "A general framework for approximate nearest subspace search," Proc. 2nd International Workshop on Subspace Methods, pp.109–116, Sept. 2009.
- [7] R. Basri, T. Hassner, and L. Zelnik-Manor, "Approximate nearest subspace search," IEEE TPAMI, vol.33, no.2, pp.266–278, Feb. 2011.
- [8] J. Hamm and D.D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," Proc. ICML, pp.376–383, 2008.
- [9] A. Edelman, T.A. Arias, and S.T. Smith, "The geometry of algorithms with orthogonality constraints," SIAM Journal on Matrix Analysis and Applications, vol.20, no.2, pp.303–353, 1998.
- [10] M.T. Harandi, M. Salzmann, S. Jayasumana, R. Hartley, and H. Li, "Expanding the family of grassmannian kernels: An embedding perspective," Proc. 13th European Conference on Computer Vision, Part VII, vol.8695, pp.408–423, Lecture Notes in Computer Science, Sept. 2014.
- [11] X. Wang, S. Atev, J. Wright, and G. Lerman, "Fast subspace search via grassmannian based hashing," Proc. ICCV, pp.2776–2783, Dec. 2013.
- [12] M. Iwamura, T. Sato, and K. Kise, "What is the most efficient way to select nearest neighbor candidates for fast approximate nearest neighbor search?," Proc. ICCV, pp.3535–3542, Dec. 2013.
- [13] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," Proc. CVPR, pp.409–415, June 2003.
- [14] 齊藤泰一, 山田博三, 山本和彦, "Jis 第 1 水準手書漢字データベース etl9 とその解析," 信学論 (D), vol.J68-D, no.4, pp.757–764, April 1985.
- [15] H. Yamada, Yamamoto, and T. Saito, "A nonlinear normalization method for handprinted kanji character recognition — line density equalization —," Pattern Recognition, vol.23, pp.1023–1029, 1990.