社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－CVIM－153（8）
2006／3／16

効率的な距離計算戦略による高次元最近傍探索の高速化

武本 浩二 †　　加藤 丈和 †　　和田 俊和 †

† 和歌山大学大学院 システム工学研究科

あ ら ま し

本論文では，高次元空間での最近傍探索の高速化について検討する．従来の最近傍探索の高速化手法には，最近傍候補の絞り込みと，距離計算の打ち切りとがある．前者は，高次元空間ではほとんど全探索になってしまい，後者は，次元数が高い場合でも，高速化が可能である．本手法では，さらに，より効率的な距離計算の打ち切りを行うために，基底を寄与率の順にソートする直交変換と，入力パターンとの距離を計算する最近傍候補の並び換えを行う．顔画像データベースを用いた実験では，従来手法よりも提案手法が高速であることを確認した．

An Accelerated High Dimensional Nearest Neighbor Search based on An Efficient Distance Computational Strategy

Koji Takemoto†, Takekazu Kato†, and Toshikazu Wada†

†Graduate School of System Engineering, Wakayama University

Abstract

In this paper, we propose an accelerated Nearest Neighbor(NN) search algorithm in high dimensional space. The methods proposed so far can be classified into two types: 1) NN candidate narrowing and 2) pruning of distance computation. In high dimensional space over 30D, while NN candidate narrowing becomes brute force search, the latter method, pruning of distance computation is still effective for acceleration. For realizing more efficient NN search in high dimensional space, we integrate these two methods. As well, orthogonal expansion of patterns ordered by contribution ratio is incorporated for efficient pruning, because the distance computation starting from the most contributed component provides good approximation of the true distance. We confirmed through extensive experiments that our method is faster than existing methods.

1　序論

最近傍探索は，ある未知入力パターンが与えられたとき，あらかじめ記憶していたプロトタイプ集合の中から最も距離の近いプロトタイプを探索する手法で，パターン認識や類似検索などの重要な基礎技術として位置づけられる．近年では，画像などのマルチメディアを用いた認識や検索などを行う機会が増え，高次元空間における最近傍探索の高速化手法の必要性が高まっている．

従来の高速化手法として，入力との距離の比較を行うプロトタイプ,（最近傍候補）を減らす手法（以下，最近傍候補の絞り込み）（[4]，[7]，[6]，[5]）と，入力との距離計算の途中で最近傍に成り得ないとわかった時点でそのプロトタイプとの距離を打ち切る手法（以下，距離計算の打ち切り）（[1]，[9]）とに大きく分けられる．これらの手法のうち，前者は高次元空間では，ほとんど全探索になってしまう．一方で，後者の距離計算の打ち切りは高次元においても安定した性能を発揮でき，また直交変換を組み合わせることにより，さらに効率化出来ることが知られている（[1]，[9]）．距離計算の打ち切りを用いて最近傍探索の高速化を行う場合，入力との距離計算を行うプロトタイプの順番が重要になってくる．従来でも，この並び換えは行われているが，任意の入力に対して有効な並び換えが行われているとは言え

ない．また，直交変換についても，基底を低周波成分の順に並び換えるだけでは，任意のデータに対して，より少ない基底で高い累積寄与率を得ることは出来ない．そこで，本論文では，前者に対しては，探索段階で入力を基にプロトタイプの並び換えを行い，後者に対しては，基底を学習段階で並び換えることにより，距離計算の打ち切りを効率的にし，高次元最近傍探索の高速化を行う手法を提案する．

2 関連研究

2.1 問題の定式化

次元数 d のベクトル $\boldsymbol{x}$ は $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_d]^T$ と表され，2 つのベクトル $\boldsymbol{x}$，$\boldsymbol{y}$ 間の距離 $d(\boldsymbol{x}, \boldsymbol{y})$ は，次式で与えられる．

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt[m]{\sum_{i=1}^{d} |\boldsymbol{x}_i - \boldsymbol{y}_i|^m} \qquad (1)$$

最近傍探索では，ある入力データ $\boldsymbol{q}$ に対するプロトタイプとの距離の大小関係のみが問題となるため，実際には，$\boldsymbol{x}$，$\boldsymbol{y}$ 間の m 乗根をとらずに比較することもできる．

プロトタイプ集合を $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_n\}$，未知入力を $\boldsymbol{q}$ とすると，最近傍探索は，次式で与えられる $\boldsymbol{p}_{\min}$ を探索する問題である．

$$\boldsymbol{p}_{\min} = \arg\min_{\boldsymbol{p}_j \in P} d(\boldsymbol{p}_j, \boldsymbol{q})^m \qquad (2)$$

全てのプロトタイプとの距離計算を行う全探索を行う場合，最近傍探索にかかるコストは，次元数 d に比例する距離計算自体にかかるコストと，プロトタイプ数 n に比例する距離計算回数を掛けた $O(dn)$ となる．

つまり，この観点から，最近傍探索の高速化手法は，大きく次の 2 つに別けられる．

- 最近傍候補の絞り込み $\cdots$ 距離の比較を行うプロトタイプを減らす．
- 距離計算の打ち切り $\cdots$ 距離の比較に用いる次元数を減らす．

また，これら 2 つの手法を組み合わせた手法も提案されている．これらの手法の概要を以下に述べる．

2.2 最近傍候補の絞り込みに基づく手法

最近傍候補の絞り込みは，距離の比較を行うプロトタイプ数を減らす手法である．最近傍候補を絞り込む方法としては，あらかじめパターン空間を分割し，探索木を構築する空間分割手法（kd-Tree[4], ANN[3], VPTree[6], GNAT[7] など）と，プロトタイプ間の相対的な距離を基に，三角不等式を用いた枝刈りを行う手法（LAESA[5], VPTree[6], GNAT[7] など）がある．

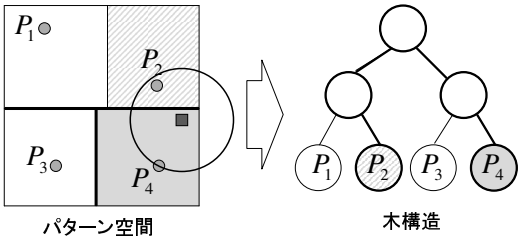


図 1: 空間分割（例：kd-tree）

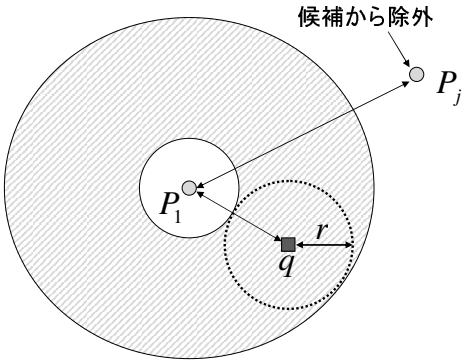


図 2: 三角不等式を用いた枝刈り（例：AESA）

前者の空間分割手法（図 1）は，あらかじめ再帰的な空間分割を繰り返し，分割された領域が各ノードに対応する木を構築する．探索時に，この木のノードに対して枝刈りを行いながら，候補となるプロトタイプを限定することにより，入力との距離計算回数を減らす．後者の三角不等式を用いた枝刈り手法（図 2）は，あらかじめプロトタイプ間の距離を計算し，ある入力パターン $\boldsymbol{q}$ から半径 r 以内にあるプロトタイプの探索を考えるとき，ベクトル $\boldsymbol{p}_1$ と $\boldsymbol{q}$ との距離 $d(\boldsymbol{p}_1, \boldsymbol{q})$ を計算すると，

$$|d(\boldsymbol{p}_1, \boldsymbol{q}) - d(\boldsymbol{p}_1, \boldsymbol{p}_j)| > r \qquad (3)$$

となるベクトル $\boldsymbol{p}_j$ は候補から除外でき，入力との距離計算回数を減らすことが出来る．

図 3 は，10 次元・20 次元・30 次元での一様乱数データ間の相対的な距離の分布を表している．この図からもわかるように，高次元空間では，どの点から見ても他の点はほぼ等距離に分布するため，パターン間の相対的な距離を基づく高速化手法は効果がない．また，パターンをベクトルと見なしたとき，高次元では，ベクトルのある成分のばらつきと，パ

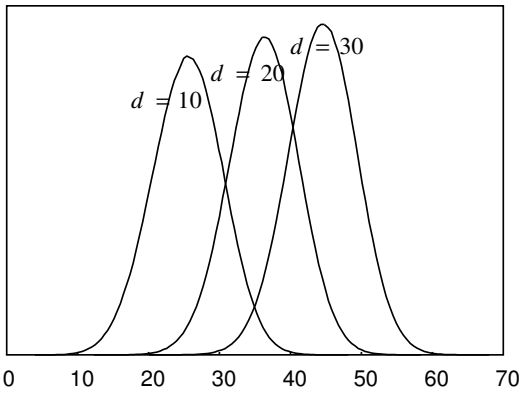


図 3: 次元の呪い（Curse of Dimensionality）
一様乱数データの相対的な距離の分布

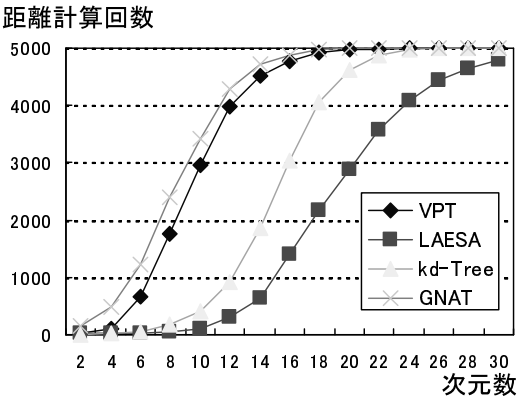


図 4: 次元数に対する距離計算回数の変化
（一様乱数データ，プロトタイプ数 5000）

ターンのばらつきとの相関関係が見出し難いため，各要素毎の空間分割は効果が少ない．

図 4 は，最近傍候補の絞り込みに基づく手法を用いて，一様乱数データの最近傍探索を行った結果である．どの手法を用いても，30 次元あたりから距離計算回数を減らすことが出来ていないことがわかる．また，単純な全探索と比較して，枝刈り判定にコストがかかっている分，かえって時間がかかってしまっている．この結果からも，最近傍候補の絞り込みだけでは高次元空間での最近傍探索の高速化は行えないことがわかる．

2.3 距離計算の打ち切りに基づく手法

距離計算の打ち切りは，プロトタイプ 1 つあたりの距離計算コストを減らす手法であり，SSDA（残差逐次検定法：Sequential Similarity Detection Algorithm）[11] で行われている．SSDA では，L_1 ノルムが用いられるが，本論文では，これを一般化し L_m ノルムで考える．

ベクトル間の距離は，

$$d(\boldsymbol{x},\boldsymbol{y}) = \sqrt[m]{\sum_{i=1}^{d} |\boldsymbol{x}_i - \boldsymbol{y}_i|^m} \tag{4}$$

で求まるが，最大の次元数 d よりも小さい $b(1 \leq b < d)$ までの基底を用いて距離計算を行うと，以下の式のように下限値が求まる．

$$\text{下限値：}\quad \underline{d}(\boldsymbol{x},\boldsymbol{y}) = \sqrt[m]{\sum_{i=1}^{b} |\boldsymbol{x}_i - \boldsymbol{y}_i|^m} \tag{5}$$

この下限値 $\underline{d}$ は基底数 b の増加に伴い単調に増加するため，正確な距離 d よりも小さく，

$$\underline{d}(\boldsymbol{x},\boldsymbol{y}) < d(\boldsymbol{x},\boldsymbol{y}) \tag{6}$$

が成り立つ．

距離計算の打ち切りは，r は閾値として，

$$\underline{d}(\boldsymbol{p}_j,\boldsymbol{q}) > r \tag{7}$$

となった時点で積和計算を打ち切り，$\boldsymbol{p}_j$ を候補から除外する手法である．最近傍探索を行う場合 r には，すでに計算された入力とプロトタイプとの距離の最小値である d^* を用いる．d^* は，次式のように表される．

$$d^* = \min_{(\boldsymbol{p}_k \in P | k<j)} d(\boldsymbol{p}_k,\boldsymbol{q}) \tag{8}$$

この距離計算の打ち切り手法は，高次元においても安定したパフォーマンスを発揮できる．

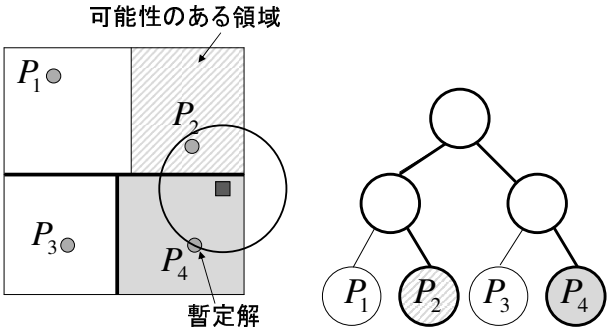
また，これらに，WHT を組み合わせた手法 [1] や主成分分析を組み合わせた手法 [9] のように，少ない基底数で，もとの情報の大部分を表現出来る直交変換を組み合わせることにより，さらに高速化できる．

しかし，距離計算の打ち切りに基づく最近傍探索の高速化を行う場合，入力との距離計算を行う順番と，効率的な直交変換が重要になるが，これらについての議論は行われていない．

2.4 Approximate Nearest Neighbor (ANN)

ANN（[3]，[2]）は，kd-tree と同様に二分木に対応付けられた空間分割による最近傍候補の絞り込みと，距離計算の打ち切りを組み合わせた手法である．

ANN は，各矩形領域に 1 つのプロトタイプが含まれるように，あらかじめパターン空間を再帰的に



分割された空間　　kd-tree

図 5: 空間分割による最近傍探索

分割する．この分割は，特長ベクトルの各要素に対し閾値を設定することに相当し，未知入力が与えれたとき，それがどの領域に含まれるかを二分探索によって求めることが出来る．

ANN による最近傍探索アルゴリズムを以下に示す（図 5 参照）．

1. 二分木探索によって，入力パターンを含む領域を探索し，その領域内のプロトタイプを暫定的な最近傍解（暫定解）とし，入力と暫定解との距離を暫定値 r とする．

2. 入力を中心・暫定値を半径とする円と重なる領域内のプロトタイプを最近傍候補とし，入力との距離を計算する．その距離が暫定値よりも小さければ，暫定値と暫定解を更新する．

3. 最近傍候補がなくなるまで１，２を繰り返す．候補が無くなれば終了．

さらに，ANN では，距離計算の際に，暫定値による距離計算の打ち切りを行なうことで距離計算自体を高速化している．

ANN が高速な理由は次の３点である．

1. 要素毎の比較による二分木探索により暫定解を高速に求める．

2. 暫定値による解候補の絞り込み．

3. 距離計算の打ち切り．

パターン空間が高次元になると，これらのうち，２の絞り込みは，ほぼ全探索となり効果がない．しかし，２により入力との距離計算を行う順番を入れ換えることにより，３の距離計算の打ち切りを効率的にしている．

3　提案手法

3.1　基本アイデア：高次元最近傍探索の高速化について

高次元空間における最近傍探索の特性を以下に挙げる．

1. 最近傍候補の絞り込みは効果がない．

2. 距離計算の打ち切りは有効．

3. 直交変換により距離計算の打ち切り効果が向上する場合がある．

4. 距離計算を行うプロトタイプの順番によって，距離計算の打ち切りの効果が変化する．

高次元での高速化のためには，距離計算の打ち切りを基本とし，その効果を最大限に発揮するために，効率のよい直交座標系へ特徴ベクトルを変換することと，効率良く打ち切りが行えるようなプロトタイプを選択することが重要である．

以下，効率的な直交変換と，プロトタイプの選択方法について議論し，高次元で高速な最近傍探索を提案する．

3.2　効率的な直交変換

特徴ベクトルの各要素１つずつが持つ情報は，次元が高くなる程少なくなるが，ベクトルの各要素の分散が偏る直交座標への変換である直交変換を行うことにより，複数の要素が持つ情報をまとめ，各基底の情報に偏りを作ることが出来る．この直交変換を用いることにより，距離計算の打ち切りの効果を向上させることが出来る．直交変換の例として，

- PCA（主成分分析：Principal Component Analysis）：分散が最大（誤差が最小）になる軸から順に選択する（図 6 参照）．少ない基底で，最も高い寄与率が得られる．
- DCT（離散コサイン変換：Discrete Cosine Transform）：周波数座標系に変換する．高速変換も可能．
- WHT（ウォルシュ・アダマール変換：Walsh Hadamard Transform）：周波数座標系に変換する．加減算のみで高速に変換可能．

などが知られている．

PCA は，最も高い累積寄与率を得ることが出来，少ない基底で多くの情報を与えることが出来るが，変換に $O(d^2)$ のコストがかかってしまい，高速化に

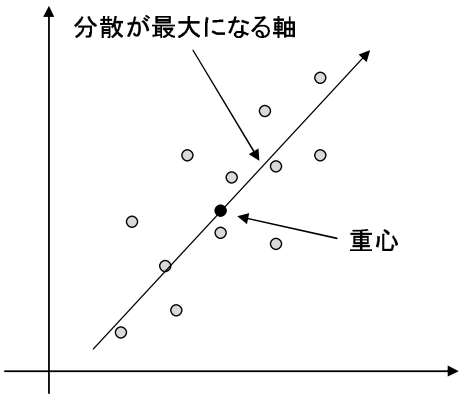


図 6: PCA の基底

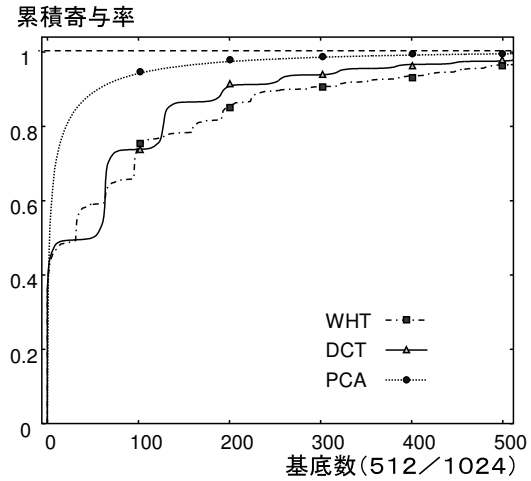


図 7: 各直交変換の累積寄与率（低周波成分順）

用いる手法としては実用的ではない．高速変換可能な直交変換として，周波数座標系に変換する DCT や，WHT などが知られている．これらは，高速アルゴリズムが存在し，$O(d \log_2 d)$ のコストで変換できる．

図 7 は，1024 次元のグレースケール画像を，これらで直交変換したときの累積寄与率である．この図から，DCT と WHT が十分な累積寄与率が得られているが，基底が低周波成分の順に並んでおり，寄与率の高い順番には並んでいないことがわかる．そこで本研究では，PCA と同様に，各基底を寄与率の降順にソートすることにより，符号化効率を向上させる．図 8 は，PCA と，基底を分散によりソートした場合の DCT と WHT の累積寄与率である．この図からも，分散でソートすることにより，累積寄与率が PCA に近づくことがわかる．

高速化手法において直交変換を用いる場合，高い累積寄与率を得ることと同様に，その変換にかかる時間も重要になるため，そのトレードオフが重要である．

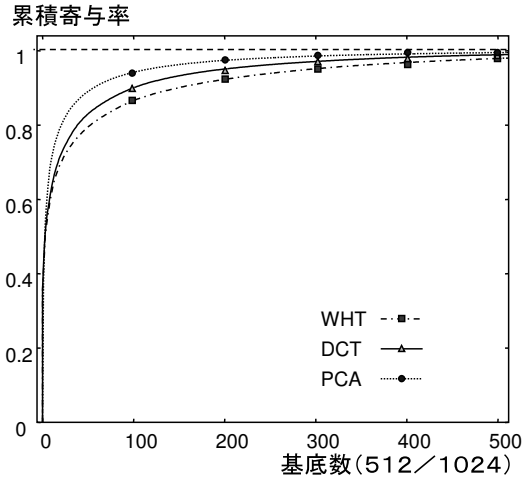
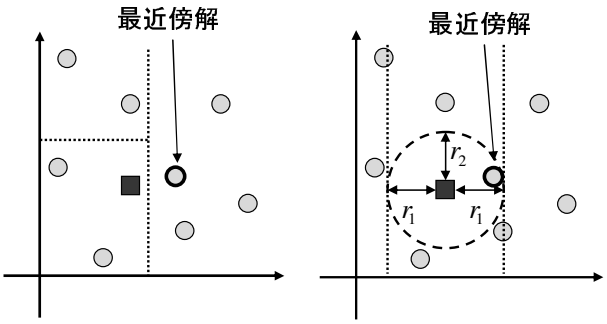


図 8: 各直交変換の累積寄与率（寄与率順）

3.3 プロトタイプの選択

未知入力から遠いプロトタイプ順に距離計算を行った場合，距離計算の打ち切りが全く行われない．このことから，未知入力に近いプロトタイプ順に距離計算を行うと，距離計算の打ち切り効果が最大限に発揮されることがわかる．実際には，未知入力に近い順番を知るためには，未知入力と全プロトタイプ間の距離を知る必要があるが，探索前にそれを知るのは不可能なため，その代替として，出来るだけ未知入力に近そうなプロトタイプ順に距離計算を行うことで，距離計算の打ち切り効果を増すようにアルゴリズムの設計を行う．本論文では，この距離計算の打ち切りが効率的に行われるような順番にプロトタイプを並び換えることを，プロトタイプの選択と呼ぶ．

実際に高次元空間で ANN による最近傍探索を行うと，全く最近傍候補の絞り込みが行われていないにもかかわらず，単純な距離計算の打ち切りを行う場合よりも高速な探索が行える．これは，ANN は，あらかじめ学習段階で空間分割により二分木を構築しておき，探索段階でこの二分木の探索を行うことにより，プロトタイプの選択を実現しており，距離計算の打ち切りにおいて探索の初期から低い暫定値が得られているためである．ANN では，学習段階で分割面を決定しているため，探索段階でのプロトタイプの選択にかかるコストが少ないという利点がある一方で，①入力が分割面付近にきたとき，選択が不安定になってしまうという欠点がある（図 9(a)）．また ANN では，②各分割面は，1 つの基底



(a)kd-tree (b) 動的二分探索

図 9: 分割面の比較（概念図）

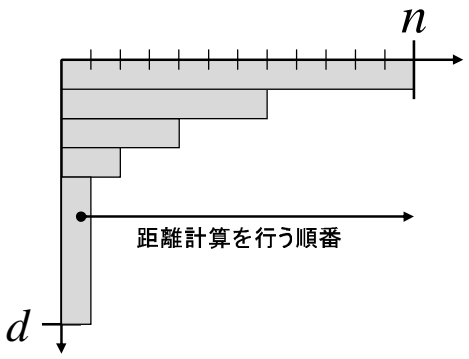


図 10: 動的二分探索

に対する情報のみを用いて決定しており，高次元では，前述の通り，各軸が持つ情報量が少ないため，効果的なプロトタイプの選択が行えない．

そこで本研究では，①に対しては，探索時に入力に対して有効な分割面を決定し，②に対しては，それまでに選択した基底から計算される下限値に基づいて分割面を決定することにより，プロトタイプの選択を安定化させる「動的二分探索」（図 9(b)）を提案する．つまり，入力とプロトタイプとの距離の下限値を求め，その下限値を基により高い次元での距離計算を続けるプロトタイプ集合を絞り込む．理想的には，中央値を基準としてプロトタイプを半分に分割するのが最適であるが，正確に中央値を求める計算はコストが高いため，本論文では，平均値を用いる．具体的な手続きを以下に示す．

1. まず最初の基底で入力と全てのプロトタイプとの距離の下限値を計算する．
2. その下限値の平均値以下のプロトタイプについて基底を増やし，新たに計算された下限値の平均値を求める．
3. ２をプロトタイプが１つになるまで繰り返す．

動的二分探索を行ったとき，より多くの基底，すなわち多くの次元を用いて下限値が計算されたプロトタイプほど入力に近いと言え，最近傍探索を行う際は，この順番に距離計算を行う（図 10）．

4 実験

4.1 実験条件

従来手法と提案手法の，高次元空間での最近傍探索のパフォーマンスを調べるために，以下のデータを用いて実験を行った．

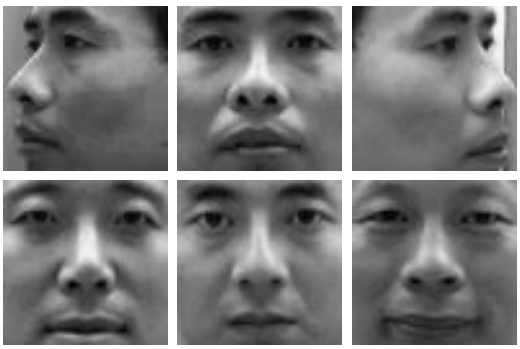


図 11: グレースケール顔画像例

- データ：グレースケール顔画像
- 次元数：1024
- プロトタイプ（学習データ）数：5000
- テスト入力データ：2000

グレースケール顔画像は CAS-PEAL のグレースケール人物画像 [8] の顔部分を抽出し，32×32 にスケーリングした画像（図 11）である．距離を L_2（ユークリッド距離）とし，テスト入力それぞれについて最近傍探索を行い，その平均を実験結果とした．直交変換は全て大浦氏の DCT[10] を使用した．

実験に用いた機材は以下の通りである．Pemtium4 PC，2.40 GHz CPU，1.0 GB memory.

4.2 各手法の比較実験

比較する手法は以下の通り．

- 手法 1： 全探索
- 手法 2： SSDA（距離計算の打ち切り）
- 手法 3： ANN
- 手法 4： SSDA+DCT
- 手法 5： ANN+DCT
- 手法 6： 提案手法（動的二分探索+DCT）

手法	使用次元数	探索時間 [ms]
1:全探索	1024.0	22.55
2:SSDA	317.7	11.81
3:ANN	281.5	10.95
4:SSDA+DCT	14.9	1.73
5:ANN+DCT	9.4	1.55
6:提案手法	9.2	1.35

表 1: 各手法の比較実験結果
使用次元数：プロトタイプ 1 つあたりの距離の比較に用いた次元数（最大値＝次元数）
探索時間は，入力が与えられてから真の最近傍解を得るまでの時間

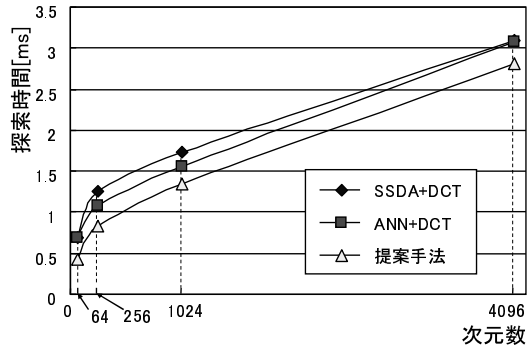


図 12: 次元数を変化させたときの探索時間の変化

以上の手法について最近傍探索実験を行った結果を表 1 に示す．DCT にかかった時間は 0.18[ms] であった．手法 4，5 と 6 の提案手法は，直交変換と距離計算の打ち切りを行っており，大変有効であることがわかる．手法 5 と提案手法はプロトタイプの選択を行っているため，それを行っていない手法 4 より高速である．手法 5 と提案手法を比較しても，使用次元数と速度の両方において，ANN に DCT を組み合わせた手法より，提案手法である動的二分探索と DCT を組み合わせたほうが有効であることがわかる．

4.3 次元数を変化させた実験

顔画像を 64 次元（8×8），256 次元（16×16），1024 次元（32×32）4096 次元（64×64）にスケーリングした画像で同様の実験を行った結果を図 12 に示す．

DCT にかかった時間はそれぞれ，0.01[ms]，0.04[ms]，0.18[ms]，0.78[ms] であった．この結果から次元が変化しても，提案手法が有効であると言える．

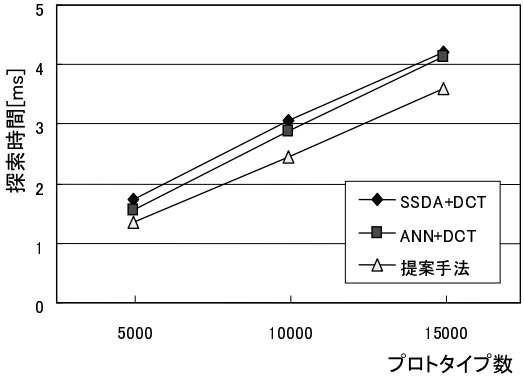


図 13: プロトタイプ数を変化させたときの探索時間の変化

4.4 プロトタイプ数を変化させた実験

プロトタイプ数を変化させ，同様の実験を行った結果を図 13 に示す．

この結果から，プロトタイプ数が増加しても，提案手法が有効であると言える．

5 まとめ

本論文では，高次元空間での最近傍探索において，距離計算の打ち切りと直交変換を用いて高速化を行い，さらに直交変換の効率化と，入力に近いと思われるプロトタイプ順に距離計算を行うプロトタイプの選択を行うことにより，高速化できること，また，そのプロトタイプの選択を安定に行う「動的二分探索」を提案した．

動的二分探索は，探索段階で入力との距離の下限値を基に分割面を決定しているため，任意の入力に対して安定な選択が可能である．

グレースケール顔画像を用いた最近傍探索実験では，以下のことを確認した．

1. 効果的な直交変換と距離計算の打ち切りの有効性：
   データに効果的な直交変換をかけることにより距離計算の打ち切りが飛躍的に向上した．

2. プロトタイプ選択の有効性：
   入力に近いと思われるプロトタイプ順に距離計算を行うことにより，距離計算の打ち切り効果が向上した．

3. 動的二分探索（提案手法）の有効性：
分割面を固定せず，入力に基づいてプロトタイプの選択を動的に行うことで，選択が安定化した．次元数・プロトタイプ数を変化させても提案手法が最も高速であった．

以上のことを実験によって確認した．

本論文では，最近傍探索を行っているが，入力と入力に k 番目に近いプロトタイプとの距離を暫定値とするのみで，k-最近傍探索が可能である．

今後の課題として，現在は平均値を用いている二分探索の基準を，より一般化し，絞り込むプロトタイプの量を調節出来るようすれば，より効率的な選択が可能である．そのためには，出来るだけ計算コストを抑えた二分探索基準が必要となる．

謝辞

本研究の一部は，文部科学省科学研究費補助金基盤研究（A）（２）16200014，及び若手研究（B)16700143 の補助を受けている．

参考文献

[1] Y.Hel-Or, H.Hel-Or, "Real-Time Pattern Matching Using Projection Kernels." IEEE Trans. Pattern Analysis and Machine Intelligence, vol.27, no.9, Sept.2005

[2] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A.Y. Wu, "An optimal algorithm for approximate nearest neighbor searching," Journal of the ACM, Vol.45, pp.891-923, 1998

[3] Library for Approximate Nearest Neighbor Searching (http://www.cs.umd.edu/mount/ANN/)

[4] J.L.Bentley: Multidimensional binary search trees used for associative searching, Communications of the ACM, Vol.18, No.9, pp.119-139, 1997

[5] LAESA : Linear Nearest-Neighbor Approximating and Eliminating Search Algorithm Mico, M.L., J. Oncina, and E. Vidal: A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. Pattern Recognition Letters (1994)15 9-17.

[6] J.K.Uhlmann, "Satisfying General Proximity/Similarity Queries with Metric Trees", Information Processing Letters, Vol40, pages175-179,1991.

[7] S.Brin, "Near neighbor search in large metric spaces." In Proc. 21th Conference on Very Large Databases (VLDB 95), pages 574-584,1995.

[8] Wen.Gao, Bo Cao, Shiguang Shan, et.al "The CAS-PEAL Large-Scale Chinese Face Datebase and Baseline Evaluation" (http://www.jdl.ac.cn/peal/index.html)

[9] 北研二，獅々堀正幹，大恵俊一郎：“ 多次元データの高速近傍検索アルゴリズム ”，情報処理学会研究報告，2003-FI-72, 2003-NL-157, pp.9-16, 2003.

[10] Takuya OOURA,General Purpose FFT (Fast Fourier/Cosine/Sine Transform) Package (http://momonga.t.u-tokyo.ac.jp/ ooura/fftman/index.html)

[11] D. I. Barnea and H. F. Silverman. A class of algorithms for fast digital image registration. IEEE Trans. on Comput., C- 21(2):179-186, 1972.