

構造化固有空間を用いた人の姿勢の高速認識

河野憲亮[†] タン ジュークイ[†] 金 亨燮[‡] 石川聖二[†]

[†] †九州工業大学

E-mail: † {kouno, etheltan, ishikawa}@ss10.cntl.kyutech.ac.jp, ‡ kim@cntl.kyutech.ac.jp

あらまし 本論文では、データベースの検索によって人の姿勢を認識する手法を提案する。固有空間法は2次元画像から3次元物体を認識する手法である。しかしこの手法では、データベースに登録されるデータ数に比例して検索時間が増大するという欠点がある。この問題を解決するために、本稿では構造化固有空間を提案する。これは固有空間にB-tree構造を適用した空間である。提案手法を人の姿勢認識に適用し、良好な結果を得た。

キーワード 姿勢認識, 固有空間, B-tree, データベース

High-speed Human Posture Recognition

Employing a Structured Eigenspace

Kensuke KOUNO[†], Joo Kooi TAN[†], Hyoungseop KIM[‡] and Seiji ISHIKAWA[†]

[†] † Kyushu Institute of Technology

E-mail: † {kouno, etheltan, ishikawa}@ss10.cntl.kyutech.ac.jp, ‡ kim@cntl.kyutech.ac.jp

Abstract In this paper, we propose a technique for retrieving a database containing a large amount of data at high-speed for the purpose of recognizing human postures. An eigenspace is a technique for recognizing a 3-D object from its 2-D image set. But it has a disadvantage that the retrieval time increases in proportion to the number of objects registered in the database. In order to overcome this difficulty, we propose a structured eigenspace. This is an eigenspace structured by B-tree and it realizes high-speed retrieval in a large amount of data in the eigenspace. An experiment was performed employing some human posture images and the proposed technique achieved satisfactory results.

Keywords structured eigenspace, eigenspace, recognition, B-tree, database

1 まえがき

近年、カメラとコンピュータを用いた人の姿勢や動作認識の研究が盛んに行われており、セキュリティ、ロボットビジョンなどへの応用が期待されている。人が、倒れたりうずくまったりした時、その姿勢や動作を監視カメラやロボット自身が自動的に検出できれば、何らかの支援を行うことができる。

そのような動作は3次元空間内で行われる。そのため、監視カメラやロボットには3次元物体認識能力が備わっていることが求められる。3次元物体の認識問題は、観察方向によらずに対象を認識できなければならないという難しさがある。これを解決するために、従来は、3次元の人モデルを画像にあてはめるという方法や、正面向き・横向きの像のみを対象とするなど観察方向を限定して姿勢・動作認識を行う方法等が主流であった。しかし前者は処理時間が膨大であること、後者は認識方向の限定が実際的でない等の問題で、いまだに実用化には程遠い状況にある。これ

を解決するために、我々は、固有空間法を用いて人の姿勢・動作認識を行う方法[3,4]を提案している。

固有空間法[2]は画像を点として記憶する方法で、姿勢や動作のあらゆる見え方の画像が、点の集合として記憶されデータベース化される。固有空間法を用いることにより、我々は3次元物体を単なる2次元画像の集合とみなすことができる。さらに、物体に関するすべての見え方は固有空間上のデータベースとして登録されるため、観察方向に寄らない物体認識が可能となる。一方、人は多様な姿勢を取り多様な動作を行うため、このデータベースは膨大となり、検索時間が実用的でなくなるという新たな問題が生じる。そこで、膨大なデータベースに対する高速な検索法が必要となる。我々は、固有空間を構造化することによりこの問題を解決する。

固有空間法では画像をn次元空間に写像し、画像の相関を距離で計算することによって類似画像の検索を行う。そのため、固有空間をデータベース化するには、高次元検索

構造が必要となる。従来、高次元検索構造を採用し、高速検索を可能とした研究として[5,6,7]などがある。[5]では、データ構造として木構造を採用しているが、この木構造は平衡木ではなく、また外部記憶装置でのデータ検索が考慮されていない。構造が平衡木となっていないため、データによっては検索時間がデータ数に比例する場合もあり、木構造のもつ利点が生かせない。また、大規模なデータベースを構築する際、そのデータサイズは膨大となるためデータを外部記憶装置に保存しなければならない。しかし、外部記憶装置は内部メモリと異なりアクセス時間が遅いため、この点を考慮する必要がある。[6]では空間を分割管理するための多次元データ管理構造として kd-tree[10]を用いているが、kd-tree はデータの次元が増加するほどアクセス時間が指数関数的に増大するという欠点がある[9]。固有空間法の性質上、データベースが膨大になるほど固有空間の次元数を高くする必要があるため、kd-tree を固有空間に適用することは難しい。このことはkd-tree の拡張型を適用している[7]にも当てはまる。

そこで本稿では、固有空間に平衡木の一種で外部記憶装置上のデータ探索に優れる B-tree[1]を適用した、構造化固有空間を提案する。本来 B-tree は多次元データ管理構造ではないが、本研究では固有空間を分割することにより B-tree の適用を可能とした。固有空間に B-tree を適用することによって固有空間が構造化され高速検索が可能となる。本研究によりデータベースの上限を大幅に引き上げることができ、知能ロボットによる人の姿勢や動作認識技術を、実用化に近づけることができる。

1 固有空間法

1.1 データベースの構築

固有空間を生成するために、まず学習に用いる画像にKL変換を施す。まず、人の姿勢 $p(p=1,2,\dots,P)$ が写されている画像 I_p を撮影する。そして I_p を TV ラスタースキャンと同じ順序で走査し、スキャンされた画素値を要素とするベクトル $\hat{x}_p = (x_1, x_2, \dots, x_N)^T$ を定義する。ここで、 N は全画素数である。

次に、画像への光学的な影響を抑えるために、ベクトル x_p を $\|x_p\|=1$ となるように次式を用いて正規化する。

$$x_p = \frac{\hat{x}_p}{\|\hat{x}_p\|} \quad (1)$$

N 行 P 列となるような行列 X を次式のように定義する。

$$X = (x_1 - c, x_2 - c, \dots, x_p - c) \quad (2)$$

ここで c は、次式で計算される画像群の平均値ベクトルである。

$$c = \frac{1}{P} \sum_{p=1}^P x_p \quad (3)$$

共分散行列 Q を行列 X を用いて次式で定義する。

$$Q = XX^T \quad (4)$$

$N \times N$ の共分散行列 Q の固有方程式

$$Qx = \lambda x \quad (5)$$

を解けば、 Q に関する N 個の固有値 $\lambda_i (i=1,2,\dots,N)$ 、固有ベクトル $e_i (i=1,2,\dots,N)$ が得られる。固有値 λ_i を降べきの順にソートし、 N より十分小さい k 個の固有値 $\lambda_1, \lambda_2, \dots, \lambda_k$ を選び、対応する固有ベクトル e_1, e_2, \dots, e_k を基底ベクトルとして k 次元固有空間

$$ES(e_1, e_2, \dots, e_k) \equiv ES \quad (6)$$

を構築すれば、ベクトル x_p は次式によって固有空間 ES 上の点 g_p に投影される。

$$g_p = (e_1, e_2, \dots, e_k)^T (x_p - c) \quad (7)$$

ここで、固有空間の構築に用いる固有ベクトルの数 k は、式(8)で与えられる累積寄与率 K により決定される。

$$K = \frac{\sum_{i=1}^k \lambda_i}{\sum_{j=1}^N \lambda_j} \leq D \quad (8)$$

累積寄与率は固有空間にベクトルを写像したときに固有空間上に残る情報の割合を表す。生成する固有空間に残す情報の割合を決定する閾値 D は適宜設定される。

上述の方法により学習画像は固有空間に投影され、固有空間上の点としてデータベースに記憶される。

1.2 未知画像の認識

未知画像 $I_{p'}$ を固有空間上に投影した座標を $g_{p'}$ とすれば、学習画像 g_p との最短距離 $d_{p'}$ は次式で与えられる。

$$d_{p'} = \min \|g_p - g_{p'}\| \quad (9)$$

$g_{p'}$ と最短距離となる座標点が存在し、かつ、ある正の値を取る閾値 ε に対して不等式

$$\|d_{p'}\| < \varepsilon \quad (10)$$

が成り立てば、未知の姿勢が $p' = p^*$ として認識される。ここで閾値 ε は実験的に定められる値である。

本研究では距離 d の定義として次式で示されるユークリッド距離を採用する。

$$d = \sqrt{\sum_{i=1}^{n_d} (x_i - q_i)^2} \quad (11)$$

ただし、 n_d は空間の次元数、 x および q は座標点である。

1.3 固有空間法の問題点

固有空間法による類似画像の検索では、式(11)を用いて最小距離の点を検索する。これは線形探索法と等価であり、平均して $n/2$ 回距離計算が行われるため、その計算量は $O(n)$ となる。ただし n は全画像数である。したがって、データベースに登録される画像に比例し探索時間も増大する。このため、大規模なデータベースに対しては検索時間が実際的でなくなるといふ問題がある。

1.4 Partial Distance Search

データベースに登録されているデータだけ式(11)による距離計算を行えば、固有空間の次元数分の積和演算を行うことになるため、計算量は膨大となる。この問題に対し、Cheng らは Partial Distance Search (PDS) [8] と呼ばれるアルゴリズムを提案している。これは式(11)に対し簡単な変更を加えたアルゴリズムである。

式(11)において、 n_{part} 次元までの部分 d_{part} を

$$d_{part} = \sum_{i=1}^{n_{part}} (x_i - q_i)^2 \quad (12)$$

とすれば、この部分 d_{part} がある正の値をもつ閾値 ε に対して

$$\varepsilon \leq d_{part} \quad (13)$$

を満たすとき、 n_{part} 次元以上の計算をやめる。

これはパターンマッチングにおける SSDA 法と酷似している。Cheng らによれば、PDS により計算量は 1/4 程度まで減少できる。本研究では式(11)を用いる際、PDS を適用する。

2 B-tree 構造

2.1 概要

B-tree とは m 分木をベースとした探索木である。 m 分探索木のうち以下の条件を満たすものを m 階の B-tree という。

- I. 根は葉であるか、または $2 \sim m$ 個の子を持つ。
- II. 根、葉以外の節は $\lceil m/2 \rceil \sim m$ 個の子を持つ。(ただし、 $\lceil x \rceil$ は x 以上の最小の整数を表す。)
- III. 根からすべての葉までの経路の長さが等しい。

B-tree では格納するデータから作られるキー (境界を表す値) が重要な意味を持ち、キーによって根や節が構築される。また、キーは大小比較することが可能なスカラー値である。B-tree では、データを持つのは葉のみで、データは葉に格納される。それ以外の節 (これを内部節 (Page) という) はキーだけを持つ。

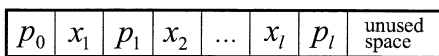


Fig. 1 Organization of a page

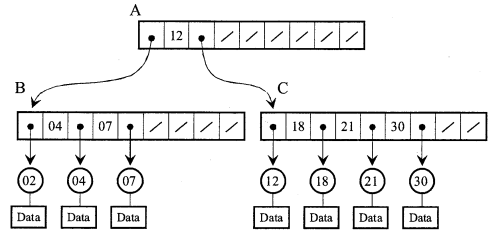


Fig. 2 An example of the B-tree whose height is 5

条件IIIから、B-tree は常にバランスが取れていることになる。B-tree の場合、節が可変個の子を持つという性質を利用して木のバランスをとる。つまり、5階の B-tree では、内部節は $3 \sim 5$ 個の子を持つことになり、この範囲内で子の数を調整して木のバランスが取れた状態に保つ。

n 個のデータを持つ B-tree の高さは、最悪の場合で $\log_{\lceil m/2 \rceil} n$ 程度、最良の場合で $\log_m n$ 程度となる。すべての節が $\lceil m/2 \rceil$ 個しか子を持っていないのが最悪の場合、すべての節が m 個の子を持っているのが最良の場合となる。いずれの場合においても B-tree の高さは $O(\log n)$ に収まる。

根や節は同じ大きさの内部節として表現される。Fig. 1 に内部節を示す。この内部節において、キー x_i は昇順に格納される。ただし、節は $\lceil m/2 \rceil \leq l \leq m$ 、根は $1 \leq l \leq m$ である。さらに、子を指し示す $l+1$ 個のポインタ p_0, p_1, \dots, p_l を保持している。Fig. 2 に5階の B-tree の例を示す。ただし、キーはデータと同じ10進数で記述された数値とする。Fig. 2 においてポインタ p_l は矢印として表現されている。大きな長方形は内部節を表し、内部にある数字はキーである。内部節の中にある斜線はキーまたはポインタが入っていないことを意味する。そして、円は葉を表し、データが関連付けられている。また、内部節の保持するキーには、隣接した右の部分木の最小値が代入される。

2.2 探索

B-tree の内部節では、キー x_i が昇順に格納され、キーは隣接した右の部分木の最小値を保持する構造をしている。このため、データを探索する際は、目的データの持つキーを探索キーとし、内部節の持つキーと比較することによって、どの部分木を辿ればよいのかを判断することができる。Fig. 2 において探索キーを 04 とおくと、まず根のキー 12 と比較し、探索キーのほうが小さいため、キー 12 の左側にあるポインタをたどり、左の節に移動する。そして、探索キー 04 は節のキー 04 以上であるため、キー 04 の右側のポインタを辿りキー 04 を持つ葉に移動する。探索キー 04 と葉の持つキー 04 は等しいため探索が成功し、葉の持つデータを引き出すことができる。

次に、B-tree における探索の計算量を求める。B-tree における探索では、根から始めて葉に到達するまでに $O(\log n)$ 個の節をたどる必要がある。また、各内部節でどの部分木

を辿るかを判断する必要があるが、この計算量は内部節のキーを順に比較すれば $O(m)$ の線形探索となるため、キーが昇順に並んでいることを利用し、2分探索を行えば $O(\log m)$ となる。一般に n の方が m よりはるかに大きいため、 m は定数とみなすことができる。したがって、B-tree における探索は常に $O(\log n)$ で実行できることになる。

3 構造化固有空間

3.1 手法

提案法は、固有空間を超立方体で表現されるピンで区分し、このピンを B-tree によって管理することにより固有空間を構造化する。

固有空間上の座標値 e_k ($k=1,2,\dots,K$) を V 個のある幅 L を持ったセクションに分割し、各セクションを固有の番号 $s_{k,v}$ ($v=0,1,\dots,V$) でラベル付ける。画像 I_p が式(7)によって固有空間の点 $\mathbf{g}_p=(g_1, g_2, \dots, g_K)$ に投影されると、 g_k ($k=1,2,\dots,K$) はいずれかのセクションに含まれるから、各軸において、セクションの固有の番号 $s_{k,v}$ ($v=0,1,\dots,V$) が与えられる。この結果、座標点 \mathbf{g}_p は次式によって K 桁 V 進数のセクション番号 S_p に変換される。

$$S_p = s_{1,v_1}s_{2,v_2}s_{3,v_3}\dots s_{K,v_K} \quad (14)$$

S_p は単なる数値であるため、B-tree のキーの条件である「比較可能なスカラー値」を満たしている。したがって、 S_p を用いて画像を B-tree に格納することができる。固有空間内における第1次元は最も固有値の大きい固有ベクトルが与えられる。これは、最も分散が大きいことを意味し、他の次元と比べて第1次元はよくデータを分けることができる。そのため、第1次元の座標値は他の次元の座標値よりも重要視されることが望ましい。式(3-1)では第1次元の座標値 g_1 が S_p の中で最上位の桁 s_{1,v_1} として与えられる。このため、第1次元の座標値 g_1 は S_p の中で支配的となり、第1次元の座標値は他の次元の座標値よりも重要であるという固有空間法の性質を反映している。

次に、構造化固有空間を用いた未知画像認識法について述べる。未知画像を I_p とすると、まず I_p を式(7)によって固有空間に投影し固有空間上の座標値 \mathbf{g}_p を得る。そして式(3-1)によって \mathbf{g}_p をセクション番号 S_p に変換する。この S_p を探索キーとして B-tree を探索することにより、 R 個の候補画像 \mathbf{g}_p ($p=1,2,\dots,R$) が得られる。そして、最後に次式を適用すれば、最も近い姿勢 $p^*=p^*$ を得ることができる。

$$d_{p^*} = \min \| \mathbf{g}_p - \mathbf{g}_{p^*} \| \quad (15)$$

$R \ll P$ となることが期待されるため、探索速度は大幅に改善される。ただし P はデータベースに登録されている全画像数である。Fig.3 に B-tree 構造を持つ2次元固有空間の例を示す。図中における e_1, e_2 は固有空間における座標軸である。また、黒丸の点は固有空間に投影された学習画像群を表しており、座標 \mathbf{g}_p に相当する。座標軸 e_1 は同一の幅 L で6分割されており、各セクションに0~5の番号がラベル付けされている。この番号が $s_{k,v}$ ($v=0,1,\dots,V$) である。座標軸 e_2 も同様に分割され、ラベル付けされている。

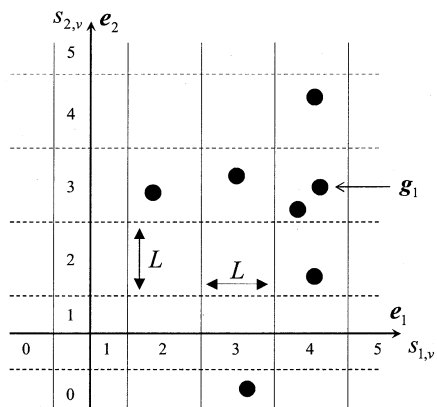


Fig. 3 A 2-D eigenspace having a B-tree structure

群を表しており、座標 \mathbf{g}_p に相当する。

Fig.3 に B-tree 構造を持つ2次元固有空間の例を示す。図中における e_1, e_2 は固有空間における座標軸である。また、黒丸の点は固有空間に投影された学習画像群を表しており、座標 \mathbf{g}_p に相当する。座標軸 e_1 は同一の幅 L で6分割されており、各セクションに0~5の番号がラベル付けされている。この番号が $s_{k,v}$ ($v=0,1,\dots,V$) である。座標軸 e_2 も同様に分割され、ラベル付けされている。

ここで、座標点 \mathbf{g}_p をセクション番号に変換することを考える。Fig.3 で、図中右上の座標点 \mathbf{g}_1 は座標軸 e_1 上の4番目のセクションに属しており、また座標軸 e_2 上で3番目のセクションに属している。したがって、座標点 \mathbf{g}_1 は2桁6進数となるセクション番号 $S_1 = 43$ が与えられる。このようにして座標点 \mathbf{g}_1 はセクション番号 S_1 に変換される。

セクション番号 S_1 は、座標軸 e_1 における4番目のセクションと、座標軸 e_2 上における3番目のセクションで囲まれた矩形領域 (ピン) を意味している。ここで、固有空間における探索範囲をこのピンの中に制限すれば、このピンの中に存在する画像 (候補画像) はデータベースに登録されている全画像数よりはるかに少なくなるため、全データベースとの全探索を行うよりも高速に探索できる。

このように、セクション番号の導入により、 n 次元固有空間を超立方体で表現されたピンに分割することができる。そして、探索範囲をその中に制限することにより高速検索が可能となる。このセクション番号をキーとして B-tree に登録することは各ピンを木構造によって関連付け、管理することに相当する。そして、セクション番号を探索キーとして B-tree を探索することにより、目的のピンへ高速にアクセスし、その中に含まれる候補画像を得ることが可能となる。ただし、候補画像の数がデータベースに登録されている画像数と同程度であると、B-tree を導入したことによる高速化は望めない。この観点から、各ピンに含まれる画像数はできるだけ少ないことが期待される。

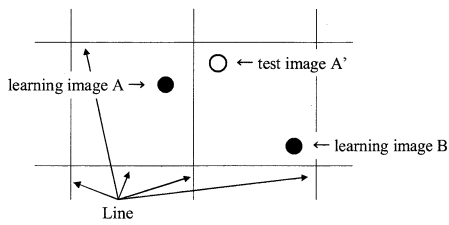


Fig. 4 Problem of border line

3.2 境界問題

固有空間を分割するために生じる問題について考察する。固有空間はセクションによって分割されるため、Fig. 4 に示すような境界問題が発生する。Fig. 4 は B-tree 構造を持った 2 次元固有空間内に、2 つの学習画像 A, B とテスト画像 A' が投影されている様子である。Line はそれぞれセクションの境界を表している。この図においてテスト画像 A' に一番近いのは学習画像 A であり、従来法による全探索では学習画像 A が正しく認識される。一方、本手法では、この 2 点の間に境界があるため学習画像 A ではなく、ビンの中で最も近い学習画像 B を選択することになる。固有空間を分割したためにこのような境界問題が生じる。

この問題は、境界付近にテスト画像が存在するときに発生するため、このような状況なるべく発生させないことが重要となる。そのためには固有空間を細かく分割することを避け、セクションの境界を少なくすべし。こうすることにより境界線付近におけるデータの密度を希薄にすることができる。しかし、固有空間を粗く分割することは各ビンに含まれる画像数の増大を招くため、高速検索を阻害することになる。このことはセクションの幅 L に大きく依存するため、最適な L を設定することが必要となる。

3.3 最適セクション幅の決定

セクションの幅 L の決定方法について述べる。 L の値をパラメータとして変化させたとき、それに伴い認識率と、ビンの中に存在する候補画像の数が変化する。

認識率 $r(L)$ は次式(16)で定義される。

$$r(L) = \frac{n_r(L)}{n_t} \quad (16)$$

ただし、 n_t はデータベース検索を行ったテスト画像の枚数、 $n_r(L)$ はその中で認識に成功した回数である。

また、候補画像の数を評価するために、候補率 $c(L)$ を次式で定義する。

$$c(L) = \frac{n_r(L)}{n} \quad (17)$$

ここで $n_r(L)$ は 1 枚のテスト画像に対してデータベース検索を行った際に得られる候補画像の数であり、 n はデータベースに登録されている全画像数である。 $n_r(L)$ は $0 \leq n_r(L) \leq n$ の関係を常に満たすため、 $c(L)$ は正規化され $0 \leq c(L) \leq 1$ となる。 $c(L)$ の値は検索速度の指標であり、 $c(L) = 1$ のとき、検索速度は従来の全探索と等しく、値が小さくなるほど検索速度は向上する。

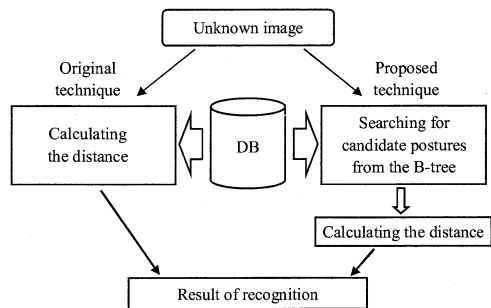


Fig. 5 Recognition process

L の値が小さいとき、固有空間を分割するビンは小さくなり、その中に含まれる候補画像は少なくなる。そのため検索速度は向上するが、境界問題が発生しやすくなるため認識率は下がる。逆に、 L の値が大きくなり、ひとつのビンに全学習画像が含まれるような場合には、境界問題は発生しないため認識率は高くなるが、検索速度は従来の全探索と変わらない。従って $r(L)$ と $c(L)$ はトレードオフの関係となり、何らかの評価値が必要となる。そこで、本研究では以下の評価値 D_{rc} を用いる。

$$D_{rc} = |r(L) - c(L)| \quad (18)$$

評価値 D_{rc} が最大となるときの L を採用すれば、認識率が高く、かつ候補率の低い L を自動的に決定することができる。また、 $r(L)$ と $c(L)$ はどちらか一方を任意に指定することもできる。もし認識率を犠牲にしても検索をリアルタイム(33msec 以下)で行いたい場合、 $c(L)$ がこの要求を満たすように L を定めることができる。

4 実験

提案手法の有効性を検証するために、人の姿勢の認識を行った。ただし、対象は人でなく人形を使用した。Fig. 6 に示す 5 種類の姿勢を人形で表現し、PC によって制御されるターンテーブル上に載せ、学習サンプルを収集した。学習用画像として 5 つの姿勢それぞれに対し、9 度おきに 360 度回転させた画像 $I'_{s,t}$ ($s = 1, 2, \dots, 5; t = 1, 2, \dots, 40$) を収集した。 s は姿勢、 t は回転角度を表す。画像の枚数は 200 枚となる。認識用として、学習用画像に対し、5.4 度ずらして収集した画像 $I''_{s,u}$ ($s = 1, 2, \dots, 5; u = 1, 2, \dots, 40$) 200 枚を用いた。 i 番目の認識用画像 $I''_{s,i}$ は $I'_{s,i}$ よりも $I'_{s,i+1}$ の方がより近い角度となるが、画像間にほとんど違いが見られないため、両画像を正解とした。実験結果を Fig. 2 に示す。評価値を最大にするセクションの幅は $L = 0.41$ であった。

$L = 0.41$ のときの各パラメータを探索時間と共に Table. 1 に示す。探索時間はテスト画像 200 枚それぞれについて得られた探索時間の平均である。Table. 1 の候補率を見ると、従来法が 100% であるのに対し、提案手法は 3.4% と計算量が抑えられていることがわかる。そして探索時間では提案

手法は約 10 倍早く、確かに高速化されている。認識率は従来法より 9%ほど低下しているが、これは境界問題によるものと考えられる。この実験よりは姿勢認識に適用することが可能であり、認識率を著しく下げることなく、検索速度の高速化が可能であることが確認された。

5 考察

未知画像の認識に必要な計算量について考察する。Fig.5 に従来の固有空間法と提案手法である構造化固有空間の認識プロセスを示す。図の左側のプロセスは従来法、右側が提案手法を示している。従来法では、データベースに対して全検索を行うため、線形探索法と等価であり、 $O(n)$ となる。ただし、 n はデータベースに含まれる全画像数である。それに対し、提案手法では、まず B-tree により候補画像を検索し、この候補画像に対して類似画像の検索を行う。B-tree による候補画像の検索は、2.2 より $O(\log n)$ の計算量となる。また、得られた R 個の候補画像に対する類似画像の検索には、 $O(R)$ の計算量がかかる。ここで、 n に対して R が十分小さければ $O(R)$ を定数とみなすことができ、提案手法の計算量は $O(\log n)$ となる。しかし、 $R \equiv n$ となると計算量は $O(n)$ となり、従来法と計算量が変わらなくなってしまふ。従って、 $R \ll n$ となるように候補画像 R を抑える必要がある。

6 結論

固有空間に B-tree を適用した構造化固有空間を用いて、人の姿勢認識を高速に行う手法を提案した。提案手法の有効性を実験的に示したが、本法はセクションの境界付近に存在するデータに対してうまく認識できないことがあるという問題が存在する。これは今後の課題である。

本研究の一部は文部科学省科学研究費補助金を受けた。

参考文献

- [1] R. Bayer, E. McCreight, "Organization and maintenance of large ordered indexes", *Acta Informatica* 1, fasc. 3, pp.173-189, 1972.
- [2] H. Murase, S. K. Nayar, "Recognition and spotting of 3D objects using parametric eigenspace representation", *Proc. of MIRU*, 2, pp.49-56, 1994.
- [3] S. Houman, M. M. Rahman, J. K. Tan, S. Ishikawa, "Representing human motions in an eigenspace based on surrounding cameras", *Proc. of ICCAS*, pp.1808-1813, 2004
- [4] M. M. Rahman, S. Ishikawa, "Robust appearance-based human action recognition", *Proc. of ICPR*, CD-ROM 4 pages, 2004.
- [5] Jau-Yuen Chen, Charles A. Bouman, Jan P. Allebach, "Fast

image database search using tree-structured VQ", *Proc. of IEEE International Conference on Image Processing*, Vol. 2, pp. 827-830, Oct., 1997.

- [6] J. McNames, "A fast nearest-neighbor algorithm based on a principal axis search tree", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-23, No. 9, pp.964-976, Sept., 2001.
- [7] T. Shibata, T. Kato, T. Wada, "K-D decision tree: An accelerated and memory efficient nearest neighbor classifier", *Proc. of 3rd IEEE International Conference on Data Mining*, pp. 641- 644, Nov., 2003.
- [8] D. Y. Cheng, A. Gersho, B. Ramamurthi, Y. Shoham, "First search algorithms for vector quantization and pattern matching", *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp.9.11.1-9.11.4, Mar., 1984.
- [9] S. A. Nene, S. K. Nayar, "A simple algorithm for nearest neighbor search in high dimensions", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.19, pp.989-1003, No.9, Sep., 1997.
- [10] J. L. Bentley, "Multidimensional binary search trees used for associative searching", *Communications of the ACM* 18, pp.509-517, Sept., 1975.

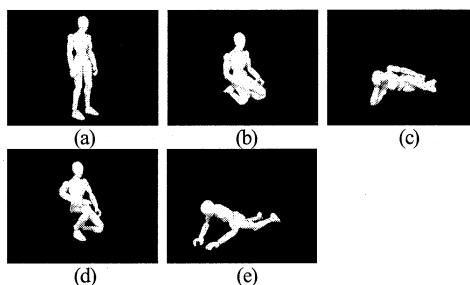


Fig. 6 Postures used in the experiment

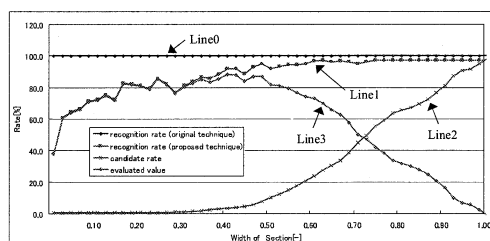


Fig. 7 Experimental results: Recognition rate with respect to the section width L

Table 1 Experimental results: Comparison of the proposed technique with the original at $L=0.41$

	Original technique	Proposed technique
Recognition rate	100 %	91.0 %
Candidate rate	100 %	3.4 %
Query time	12 msec	1.2 msec