

特徴ベクトルの簡略離散表現による モーション検索高速化手法

高原 健輔^{1,a)} Natapon Pantuwong² 吉川 毅¹ 野中 秀俊¹ 杉本 雅則¹

概要: 本稿では、モーションの各フレームを整数値を要素とする特徴ベクトルで表現することにより、Dynamic Time Warping の計算を高速化し検索時間を短縮する手法を提案する。フレームにおける人型スケルトンの姿勢を表す特徴ベクトルに対して主成分分析を行い、各主成分の次元に対応する要素を2ビット量子化する。特徴ベクトルで表現された各フレーム間の類似度行列を生成することで、DTW の計算時間を高速化する。評価実験では、検索性能に関する予備実験を通して最適な特徴ベクトルの次元を決定した。次に、提案手法と既存手法との比較実験を行った。その結果、従来の DTW よりも 30 倍以上高速であり、かつ k-d tree を用いる [1] の方法とほぼ同程度の正解率と検索時間を示すことを確認した。

A Rapid Motion Retrieval Technique using Simple and Discrete Representation of Feature Vector

TAKAHARA KENSUKE^{1,a)} NATAPON PANTUWONG² YOSHIKAWA TAKESHI¹ NONAKA HIDETOSHI¹
SUGIMOTO MASANORI¹

Abstract: In this paper, we propose a rapid motion retrieval technique using Dynamic Time Warping. Frames of motions are represented by feature vectors whose elements are integer values. The number of the feature vector dimension is reduced by using the Principal Component Analysis method and values of elements in the vector are quantized into two bits. The similarity matrix between frames of motions represented by the feature vectors is generated for rapid calculation of Dynamic Time Warping. Preliminary experiments are conducted to find the optimum dimension number of the feature vector by evaluating the motion retrieval performance. Comparative experiments with existing methods have proved that our proposed technique can complete retrieval tasks more than 30 times faster than the traditional Dynamic Time Warping method and shown almost the same level of precision and rapid calculation time as the method described in [1] using the k-d tree algorithm.

1. 序論

近年のコンピュータの性能向上と技術の発展によって、高品質の3次元コンピュータグラフィックス(以下3DCG)アニメーションが身近な存在になっている。また商用、あるいはフリーのコンピュータアニメーションソフトウェアの入手も容易になりつつある。

現在発売されている3DCGアニメーション制作ソフトウェアを使用して、キャラクターのアニメーションを制作

するためには大量の時間と労力が必要である。キャラクターのモーション編集には、一つ一つの部位の動作を逐次的に決定しなければならない。よって膨大な編集作業が必要となり、プロフェッショナルなコンピュータアニメーターにとっても簡単にアニメーション制作が行えるわけではない。

現在では、事前に高価な3DモーションキャプチャデバイスやKinectなどにより記録されたモーションや、他人が編集したダンスモーションなどがインターネット上で入手可能となっている。アニメーション制作の初心者でも、あらかじめ録画・制作されたモーションを利用することにより、煩雑な作業をすることなく自然なアニメーションを

¹ 北海道大学大学院情報科学研究科

² King Mongkut's Institute of Technology Ladkrabang

^{a)} takahara@main.ist.hokudai.ac.jp

制作できるようになりつつある。

モーションを再利用するための研究は、これまでに多数なされている。そのうちの 하나가モーション検索である。モーションは高次元データであり計算コストを低くするためには、次元を削減しつつ正解率の高い検索を実現できる特徴量の抽出が必要である。また、モーションは時系列データなので、他のモーションと1対1のフレーム間でのアラインメントを取って比較する方法が取られることが多く、よって計算コストが高くなりやすい。

本稿で提案する手法は、まず最初にモーション中の各フレームから抽出した実数値の特徴ベクトルに対し、主成分分析を適用することより次元削減を行う。次に、各主成分に対応する要素の値を0, 1, 2, 3の整数で2ビット量子化することで、各フレームを整数値の特徴ベクトルで表現する。フレーム間の距離は、各フレームに対応する特徴ベクトルを用いて生成される類似度行列より高速に求めることができる。モーション検索で用いる Dynamic Time Warping(以下 DTW) は、各フレーム間の距離の総和の最小値を求めるアルゴリズムである。よって、提案手法によりフレーム間の距離計算の高速化することで、モーション検索の高速化が可能となる。

提案手法の実装に当たり、各フレームに対応する実数値の特徴ベクトルに対して主成分分析を適用し、次元削減とモーション検索性能の関係を予備実験で調査した。提案手法の評価実験では、予備実験で最適と判断した次元数により各フレームを整数値の特徴ベクトルで表現した。既存手法との比較の結果、提案手法は従来の DTW の約 30 倍の高速化が可能であること、k-d tree を用いる Forbes らの手法 [1] とほぼ同等の正解率と計算時間でモーション検索が可能であることを、それぞれ確認した。

2. 関連研究

モーション取得に関する手法は現在までに多数提案されている。モーションは高次元の時系列データであるため、主に次元削減が目的の特徴量抽出や時系列データの効果的な比較方法が多い。

モーション比較には DTW[2] がよく用いられる。DTW は時系列データに対して適切なアラインメントを取り比較を行うが、既に述べたとおり計算量が大きいことが欠点である。モーションは DTW で扱うには高次元であるために、モーションの特性を失わず DTW に適用できるようにモーションの特徴ベクトルを次元削減する必要がある。Forbes[1] はモーションの各フレームに対して重み付き主成分分析を用いてフレームの次元を削減した。射影後の主成分で再構成されたフレームを k-d tree を用いてクラスタリングし、フレーム間の比較のためのインデックスを用意することで DTW を高速化した。Hsu[3] らの手法は事前にモーションの速度や加速度を計算することにより、動

きが少ないフレームを削除することでモーションを削減する。さらに、動きの特性を失わないようにモーションを圧縮することで DTW の計算時間を短縮する手法を提案した。Zhu[4] はクォータニオンで表される関節の回転情報に対して、K-SVD アルゴリズムを用いて辞書学習を行った。そして、モーション情報の分解を行い辞書データを用いたモーション圧縮・検索手法を提案した。Sun[5] らの手法は、モーションを Self-Similarity Matrix の形にして低ランクの部分空間に分解することによりフレームが持つ情報を維持しつつ次元削減する。Krüger[6] らの手法は、データベース内の全モーションを k-d tree に格納する。入力モーションの各フレームを k 個の近傍と比較することで、DTW の計算時間を短縮した。

DTW を使用せずに、独自の手法でモーションを部分的に分割して検索を行う手法も提案されている。Huang[7] はつま先・指先などの先端部分が親関節(膝・肘など)から見て半球の表面上を動いていると想定し、少数のパラメータでモーションを表現する。この表現を用い、パラメータの比較のみでモーション検索を行う手法を提案した。Qi[8] は K 平均法によりデータベースから多数のキーポーズを生成し、データベース内のモーションをフレーム単位で全てクラスタリングする。混合ガウスモデルを使用した最尤推定により、入力されたモーションを全ていずれかのクラスに当てはめることでモーション比較を単純化した。Kapadia[9] はモーションに対して部位間の幾何学的な距離特徴から速度・加速度を用いた動きの力強さ・繊細さを表す様々な特徴量を文字として扱う。その後 Trie 木に格納し、検索の高速化を図る。Chao[10] は各部位の移動の軌跡を CG のレンダリングに用いられる球面調和関数で球の集合体に近似する。モーションを少数のパラメータで表し、小さい動きのノイズに対して頑健な検索を可能にしている。Zhou[11] は Sparse Representation を用い、ピラミッド状の 2 次元ヒストグラムに分割して各ヒストグラムを比較することによるモーション比較手法を提案した。

次元削減以外では、モーションの特性を反映した特徴量抽出の研究も行われている。Müller[12] は部位間の距離、関節の角度、3次元空間を分割する平面を用いた布尔型特徴量など、いくつかの特徴量を提案した。そして、DTW を用いてモーション取得の実験を行い各特徴量の有効性を検証している。Chen[13] は身体全体ではなく部分的なモーションを取得するために有効な特徴量の選択と、その重み付けの手法を提案した。

データベースから録画されたモーションを取得するための入力インタフェースに関する研究もある。Choi[14] は 3次元モーションを平面の棒人間に書き起こし、ユーザが棒人間を書き込むだけで必要なモーションを取得できるインタフェースを提案した。漫画等で見られるような動きを表す曲線や棒人間の目の位置などで身体の向きを表し、平

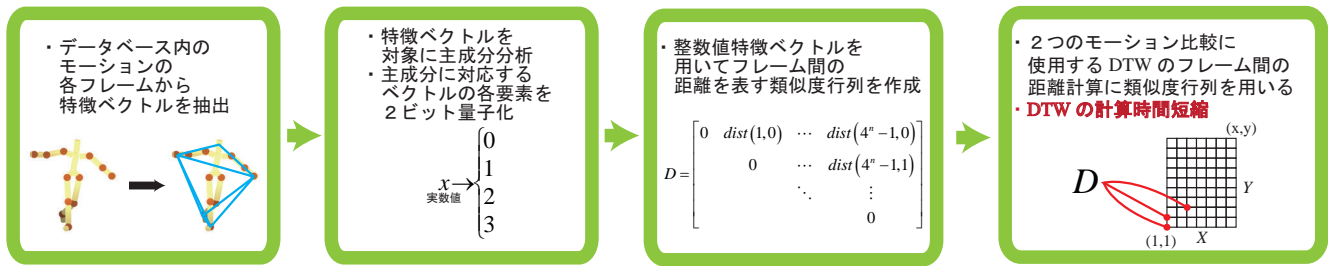


図 1 提案手法の概要

面上で 3 次元の情報を補間している。Oshita[15] はマルチタッチ入力を用いたキャラクター操作によるアニメーション生成手法を提案している。パンチ、キック、ジャンプ等それぞれのアクションに 1 種類ずつの学習済みモデルを用意して、ユーザの入力とそれぞれの学習済みモデルから得たキャラクタの動いた軌跡の曲線とを比較することでアクションを選択し、自然で連続的なアニメーション生成を行っている。

DTW の計算量を減らす手法も、これまでに提案されている。Sakoe[16] らは 2 つの時系列データのアライメントをとる際に距離を計算する範囲を有効な部分に限定することより、正解率を落とさず DTW を高速化する手法を提案した。Keogh[17] らは、時系列データを比較するための前処理で下限値を計算し複数回の比較の検索時間を速める手法を提案した。

3. 提案手法

提案手法では、モーション中の各フレームに対して整数値の特徴ベクトルを生成し、DTW の距離関数の計算を高速化することでモーション検索の実行時間を短縮する。提案手法の概要は図 1 に示す。モーションの各フレームについて、スケルトン表現のキャラクタの手足などの部位間の距離を基に 17 次元の特徴ベクトルを抽出する。その特徴ベクトルに対して主成分分析を行い、各主成分に対して設定された閾値を用いて特徴ベクトルの各要素を 0, 1, 2, 3 の 4 値で表す。2 ビットで表された新たな特徴ベクトルを用いて、フレーム間の距離を表す類似度行列を作成する。DTW のフレーム間の距離計算に類似度行列を用いることで、モーション検索の計算を高速化する。

3.1 特徴量抽出

提案する特徴量抽出手法は、多次元の関節の角度情報(実数値のベクトル)が時系列順に並んだモーションデータファイルを、整数値のベクトルが時系列順に並んだデータに変換する。以下に変換手順を示す。

モーションデータファイルには、関節の親子構造を含めた角度情報がフォワードキネマティクスで記述されている。インターネット上で公開されている CMU のデータ

ベース [18] のモーションデータは、1 フレームにつき 32 箇所の関節部位の 3 次元回転情報(計 96 次元)を持つ。高速なモーション比較をするためには次元削減を行いつつ、モーション間の類似度が正確に計測できる特徴ベクトルを抽出する必要がある。今回は Müller[12] らの手法を参考に、モーションの性質を表す特徴ベクトルとして身体の部位間の距離を用いた。

全モーションの各フレームの距離特徴量に対して、主成分分析を適用することにより特徴ベクトルの次元数を削減する。これにより、寄与率が上位 n 個の主成分からなる特徴ベクトルによって各フレームが表現される。主成分にかかる係数は以下の式により 2 ビットの整数に変換され、特徴ベクトルの要素の値として与えられる。 n を特徴ベクトルの次元数、 i を寄与率上位 i 番目の主成分 ($1 \leq i \leq n$)、 a_{ji} をフレーム j の主成分 i に対応する係数、 μ_i を主成分 i の係数の平均、 σ_i をその標準偏差とする。フレーム j を表現する特徴ベクトルを $\mathbf{g}_j = (g_{j1}, \dots, g_{jn})$ とすると、主成分 i に対するフレーム j の要素 g_{ji} は式 (1) のように計算される。

$$g_{ji} = \begin{cases} 0 & a_{ji} < \mu_i - \sigma_i \\ 1 & \mu_i - \sigma_i < a_{ji} < \mu_i \\ 2 & \mu_i < a_{ji} < \mu_i + \sigma_i \\ 3 & a_{ji} > \mu_i + \sigma_i \end{cases} \quad (1)$$

g_{ji} は 2 ビット量子化され、0, 1, 2, 3 の 4 値のいずれかになる。以下の式 (2) により特徴ベクトル \mathbf{g}_j を整数値特徴量 m_j に変換することで、モーション中の全てのフレームは 0 から $4^n - 1$ までのいずれかの整数値に対応付けられる。この整数値特徴量は、3.2 節に示す類似度行列のインデックスとして用いられる。

$$m_j = \sum_{i=1}^n g_{ji} \times 4^{i-1} \quad (2)$$

3.2 フレーム比較に用いる類似度行列の作成

前節で示した特徴ベクトルを用い、DTW の計算時間を短縮するために類似度行列を生成する。主成分 i に対応する固有ベクトルの寄与率を c_i 、フレーム j および k を表す特徴ベクトル $\mathbf{g}_j, \mathbf{g}_k$ の主成分 i に対応する要素をそれぞれ

g_{ji} , g_{ki} とする. 式 (2) より g_j , g_k は整数特徴量 m_j , m_k に対応する. 式 (3)(4) に示すように, g_{ji} と g_{ki} の差の絶対値に c_i で重みづけした値を $weightedDiff$ で求め, その総和をフレーム j , k 間の距離 $dist$ と定義する.

$$dist(m_j, m_k) = \sum_{i=1}^n weightedDiff(g_{ji}, g_{ki}) \quad (3)$$

$$weightedDiff(g_{ji}, g_{ki}) = \begin{cases} 0 & |g_{ji} - g_{ki}| = 0 \\ c_i & |g_{ji} - g_{ki}| = 1 \\ c_i \times 4 & |g_{ji} - g_{ki}| = 2 \\ c_i \times 4^2 & |g_{ji} - g_{ki}| = 3 \end{cases} \quad (4)$$

各フレームは n 次元の特徴ベクトルで表現されているので, 式 (2) に示した通り $2 \times n$ ビットの非負整数 (すなわち 0 から $4^n - 1$) に対応付けられる. DTW の計算時間を短縮するために, 各フレーム間の距離を式 (3) と式 (4) を用いて 4^n 次正方形行列に格納する. この行列 D を類似度行列と呼ぶ. 類似度行列 D の対角成分は同じ特徴ベクトルに対応するフレーム間距離となるため, 値は 0 になる. 類似度行列 D は対称行列となるため, 式 (5) に示すように上三角行列または下三角行列で表現できる. 各フレームの特徴ベクトルで計算される整数値特徴量 m を行と列に指定すれば, それらの距離が求まる. 式 (5) で例えば, フレーム j , k 間の距離は, m_j 行 m_k 列の要素である. モーション間の比較では, 次節で示すように, 対応付けられた全てのフレーム間の距離を加算するだけでよいので計算時間が短縮できる.

$$D = \begin{bmatrix} 0 & dist(1,0) & dist(2,0) & \cdots & dist(4^n-2,0) & dist(4^n-1,0) \\ & 0 & dist(2,1) & \cdots & dist(4^n-2,1) & dist(4^n-1,1) \\ & & \ddots & \cdots & & \vdots \\ & & & & 0 & dist(4^n-1,4^n-2) \\ & & & & & 0 \end{bmatrix} \quad (5)$$

3.3 類似度行列を用いた DTW

上記の類似度行列を用いた DTW によるモーション間距離の計算は, 以下のように実行される. 2 つのモーションのフレーム数を x と y とする.

Step 1 $x \times y$ 行列を用意し, 2 つのモーションの各フレーム間の距離を式 (5) の類似度行列から求めて格納する.

Step 2 動的計画法により行列の (1,1) から (x,y) までの距離の総和が最も低くなるパスを求める.

DTW では, 上記の Step 1 の計算が占める割合が高い. 提案手法では各フレームを既に整数値特徴ベクトルに変換しており, かつ特徴ベクトル間の距離を格納した類似度行列 D を作成済みである. DTW における各フレーム間の距離計算では図 2 のようにインデックスが用意されているため, Step 1 の計算時間が大幅に短縮できる.

4. 実験

提案手法の性能を評価するための実験を行った. モーション検索の正解率と計算時間を既存手法と比較する実験を行い, 提案手法の有効性を確認した. なお, 本手法は Windows 7 上で C++ で実装された. 実験では, AeroStream RM5J-B41/S (Intel Core i5, 3.10 GHz, 4.00 GB memory) を用いた.

4.1 予備実験

特徴ベクトルの最適な次元数を選ぶに当たり, 次元数がモーション検索性能にどのような影響を与えるかを調査する予備実験を行った.

4.1.1 予備実験詳細

CMU のモーションデータベース [18] から複数のモーションを選択して独自のデータベースを作成する. 5 種類の動作 (jump, walk, run, dance, kick) を選択し, 各モーションにその動作のクラスをラベル付けしておく. このデータベースに対して 3 分の 1 を入力モーション, 残りを参照モーションとする 3 分割の交差確認法でモーション検索を行った. 入力モーションと同じクラスの参照モーションが結果として出力されれば正解とした. クラスの正解率と検索に要した時間を計測した.

予備実験では, 次元数を 1 から 7 まで変化させた. 用いたデータベース中のモーション数は 114 個 (全時間長 794 秒) である.

4.1.2 予備実験結果と考察

図 3 は, 次元数と正解率の関係を示している. 次元数 4 以上で正解率が 90% を超えた. 提案手法の性質上, 次元数が多いほど正解率は高くなるが, 類似度行列の大きさも次元数の 4^2 倍で増大する. 次元数の影響は検索の計算時間を示した図 4 にも現れている. 次元数が 1 から 6 までは次元数の増大とともに計算時間がなだらかに増大しているが, 7 のとき計算時間が大きく増えている. この原因は, 類似度行列のサイズが大き (4^7 次正方形行列) ので行列を 4 つに分割して実装したためである. 図 5 に示す累積寄与率は, 次元数 2 以降は一定の差分で増加している. また, 累積寄与率が 70% 以上あれば, 提案手法は 80% 以上の正解率を示すことを確認した.

以上の予備実験から, 次元数 6 で高速かつ高正解率でモーション検索が出来ると判断し, 以降の提案手法と既存手法との比較実験は次元数を 6 と設定して行った.

4.2 比較実験

3 章で説明した提案手法と, 既存の 2 種類のモーション検索手法を比較した.

4.2.1 比較実験詳細

既存手法の 1 つ目は, Müller らの手法 [12] を用いて実数

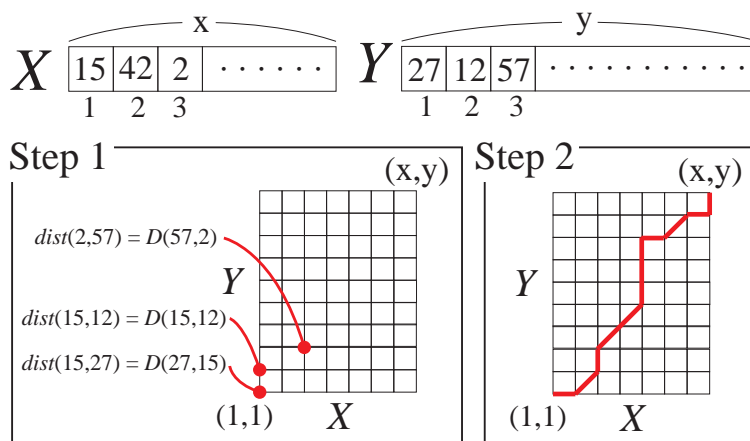


図 2 提案手法における DTW

値の 17 次元の特徴ベクトルを構成し、従来の DTW よりモーション検索を行う手法である。2 つ目は、Forbes[1] による重み付き主成分分析と k-d tree を用いる手法である。

予備実験と同様の手法で CMU[18] から作成した独自のモーションデータベースに対し 3 分割の交差確認法で実験を行い、クラスの合致率による正解率と検索時間を計測した。データベース内のモーションが 114 個 (全体の長さが 794 秒) と 162 個 (1278 秒) の 2 ケースで実験した。

4.2.2 比較実験結果と考察

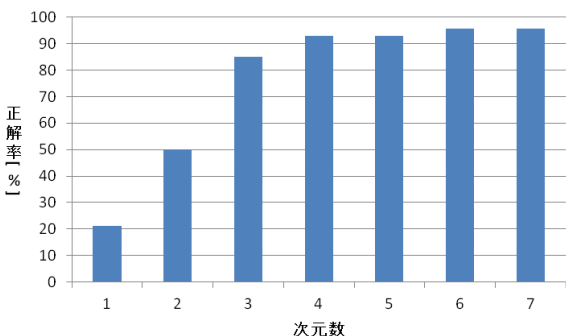


図 3 モーション検索のクラス正解率

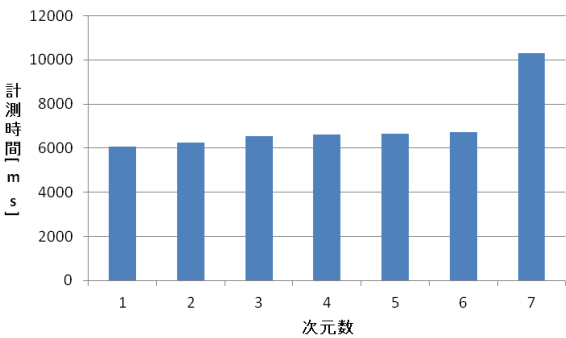


図 4 モーション検索の計算時間

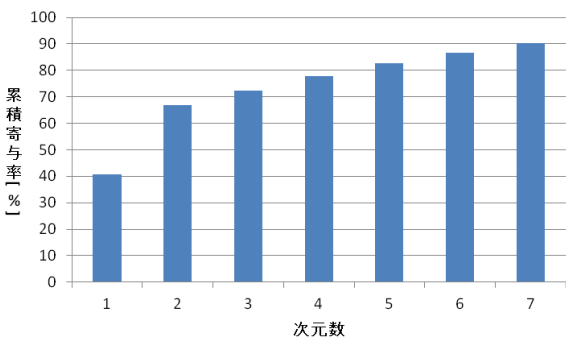


図 5 射影された主成分の累積寄与率

表 1 114 個 (794 秒) の結果

	提案手法	DTW	Forbes
時間 (秒)	7.768	240.579	6.869
正解率 (%)	94.737	97.368	95.614

表 2 162 個 (1278 秒) の結果

	提案手法	DTW	Forbes
時間 (秒)	17.914	624.199	17.128
正解率 (%)	95.062	96.296	94.444

表 1 が 114 個、表 2 が 162 個のモーションを使用した実験の結果である。

2 つのケースで、従来の DTW よりも提案手法と Forbes らの手法が約 30 倍計算時間が速いことが分かる。モーション数が少ない場合、提案手法と Forbes らの手法との計算時間の差は大きいですが、モーション数が増えると計算時間の差がやや小さくなる。提案手法では全モーションの各フレームを整数値にすることにより類似度行列の要素を抽出するので、距離計算時間への影響は小さい。一方、Forbes らの手法は k-d tree を用いて特徴量抽出・類似度行列作成を行うので、データ数が多いほど k-d tree が大きくなり、提案手法よりも検索時間への影響が大きい。

正解率に関しては提案手法、Forbes らの手法は主成分分析を用いて次元削減しているにも関わらず、従来の DTW とほぼ同程度の正解率を示すことが確認された。モーション数が多い場合に、Forbes らの手法よりも提案手法の正解

率が若干高くなった。提案手法と Forbes らの手法は全訓練モーションの各フレームをクラスタリングした後に、クラスタごとの距離をあらかじめ類似度行列に格納する。そして、DTW のフレーム間の距離の計算に類似度行列を用いることによって計算時間を短縮する点で共通している。両者の本質的な相違点は採用したクラスタリング手法であり、提案手法では主成分分析と 2 ビット量子化を用いているのに対して、Forbes の手法では重み付主成分分析と k-d tree を用いている。データベースが大きいとより精度の高い主成分分析が行えるため、提案手法の方が正解率が高い。今後はデータセットの大きさやクラス数など多様な条件下で実験することで、2 つの手法の詳細な性能を明らかにする必要がある。

5. 結論と今後の課題

本研究では、モーションにおける各フレームの特徴ベクトルを主成分分析し、2 ビット量子化を通して整数値に置き換え類似度行列を作成することにより、モーション検索の正解率を維持しつつ高速化する手法を提案し実装した。提案手法の有効性の確認のために、2 種類の既存手法との比較実験を大小 2 つのデータセットで行った。簡略かつ離散化された特徴ベクトルによるフレーム表現でありながら、提案手法は既存手法に比肩しうる性能を示すことが確認できた。今後はより多様なデータセットで実験を行い、提案手法の利点と限界、さらに実験結果に基づく改良を行う必要がある。また、提案手法に適した特徴量の抽出、特徴ベクトルの分布に基づく類似度行列の圧縮と計算時間の更なる短縮などが今後の課題として挙げられる。

参考文献

- [1] Forbes, K., & Fiume, E. (2005). An efficient search algorithm for motion data using weighted PCA. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (pp. 67-76).
- [2] Rabiner, L. R., & Juang, B. H. (1993). *Fundamentals of speech recognition* (Vol. 14).
- [3] Hsu, E., da Silva, M., & Popović, J. (2007). Guided time warping for motion editing. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (pp. 45-52).
- [4] Zhu, M., Sun, H., & Deng, Z. (2012). Quaternion space sparse decomposition for motion compression and retrieval. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (pp. 183-192).
- [5] Sun, C., Junejo, I., & Foroosh, H. (2011). Motion Retrieval Using Low - Rank Subspace Decomposition of Motion Volume. In *Computer Graphics Forum* (Vol. 30, Issue. 7, pp. 1953-1962).
- [6] Krüger, B., Tautges, J., Weber, A., & Zinke, A. (2010). Fast local and global similarity searches in large motion capture databases. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (pp. 1-10).
- [7] Huang, T., Liu, H., & Ding, G. (2012). Motion retrieval based on kinetic features in large motion database. In *Proceedings of the 14th ACM international conference on Multimodal interaction* (pp. 209-216).
- [8] Qi, T., Feng, Y., Xiao, J., Zhuang, Y., Yang, X., & Zhang, J. (2013). A semantic feature for human motion retrieval. *Computer Animation and Virtual Worlds* (Vol. 24, Issue. 3-4, pp. 399-407).
- [9] Kapadia, M., Chiang, I. K., Thomas, T., Badler, N. I., & Kider Jr, J. T. (2013). Efficient motion retrieval in large motion databases. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (pp. 19-28).
- [10] Chao, M. W., Lin, C. H., Assa, J., & Lee, T. Y. (2012). Human motion retrieval from hand-drawn sketch. *Visualization and Computer Graphics, IEEE Transactions on* (Vol.18, Issue. 5, pp. 729-740).
- [11] Zhou, L., Lu, Z., Leung, H., & Shang, L. (2014). Spatial temporal pyramid matching using temporal sparse representation for human motion retrieval. *The Visual Computer* (Vol. 30, Issue. 6-8, pp. 845-854).
- [12] Müller, M., & Röder, T. (2006, September). Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (pp. 137-146).
- [13] Chen, S., Sun, Z., Li, Y., & Li, Q. (2012, November). Partial similarity human motion retrieval based on relative geometry features. In *Digital Home (ICDH), 2012 Fourth International Conference* (pp. 298-303).
- [14] Choi, M. G., Yang, K., Igarashi, T., Mitani, J., & Lee, J. (2012, September). Retrieval and visualization of human motion data via stick figures. In *Computer Graphics Forum* (Vol. 31, Issue. 7, pp. 2057-2065).
- [15] Oshita, M. (2012). Multi-Touch Interface for Character Motion Control Using Example-Based Posture Synthesis, 20th International Conference on Computer Graphics. *Visualization and Computer Vision 2012* (pp. 213-222).
- [16] Sakoe, H. & Chiba, S. (1978), Dynamic programming algorithm optimization for spoken word recognition. *Trans. on ASSP* (Vol. 26, Issue. 1, pp. 43-49).
- [17] Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and information systems* (Vol. 7, Issue. 3, pp. 358-386).
- [18] C. G. Lab. CMU Graphics Lab Motion Capture Database. <http://mocap.cs.cmu.edu/>