To cite this version :

An Efficient Approach for Human Motion Data Mining Based on Curves Matching

Van-Hanh Nguyen[1], Frederic Merienne[2], and Jean-Luc Martinez[3]

[1] Arts et Métier ParisTech - Le2i - CNRS
3 Rue Thomas Dumorey, 71100 Chalon Sur Saone, France
hanhnv@gmail.com
[2] frederic.Merienne@ensam.eu
[3] jean-luc.martinez@cluny.ensam.eu

Abstract. In this paper, we present a novel and efficient approach to retrieve human motion capture data as used in data-driven computer games, animated movies and special effects in the aim of finding a specific motion. From the kinematic chain model, the human motion capture data is transformed to a spatial-temporal invariance representation called the motion feature representation, in which each segment of kinematic chain model is represented by an angle between itself and the root segment. We treat the human motion as a cluster of curves of angle. In the aim of finding a human motion capture data in a very large database, we propose a novel lower bounding distance called LB_Keogh_Lowe to speed up similarity search. In order to reduce the computational cost, we employ techniques to simplify the curves length of both the envelopes curves and the query data. The similarity between two human motions is measured by applying the constrained Dynamic Time Warping. We carry out an experimental analysis with various real motion capture dataset. The results demonstrate the efficiency of our approach in the context of the human motion capture data and the potentiality to apply it in others contexts of the time-series data retrieval.

1 Introduction

Nowadays, motion capturing plays an important role in computer games, computer-animated movies or video games as a tool for synthesizing realistic motion sequences. In general, by using optical devices, trajectories of moving markers attached to the human body are recorded as 3D motion data. The huge quantity of motion capture dataset is very expensive and time consuming due to the high cost of recording equipment, high computational power and the requirement of manual post-processing operations. Therefore, it has been suggested reusing the motion capture data by modifying and adapting the existing motion clips via editing and morphing techniques [1][2][3]. Hence, there is a strong need to develop an efficient method to retrieve a motion data among others.

In practice, content-based approach is one of the most efficient methods to enable an automatic extraction of a user-specified motion. The crucial point of

this approach is to define the feature of motion content utilized to qualify the motion. Some existing methods consider the motion key-frame as the feature but cannot handle the spatial-temporal variation of motion [4]. In this paper, we propose a novel approach consisting in transforming the motion capture data to a spatial invariance space called motion feature representation. For that, we use the kinematic chain model to represent the motion capture date. In the spatial invariance space, each segment of kinematic chain model is represented by an angle between itself and the root segment. Thus, we treat the transformed motion as a cluster of curves of all the angles describing the kinematic motion. The distance between two motions is defined as the similarity between the curves of the motions. Hence, we conduct the motion data mining as the shape matching problem between the curves. In the aim of finding a human motion capture data in a very large database, we propose techniques to speed up similarity search based on bounding envelopes curves. In order to reduce the computational cost, we employ techniques to simplify the curves length of both the envelopes curves and the query data. The similarity between two human motions is measured by applying the constrained Dynamic Time Warping technique on the curves of their angles. Our contributions can be summarized as following:

- We introduce the spatial-temporal invariance representation for human motion capture data and conduct the human motion data mining to the curves matching problem.
- We present a novel lower bounding distance called LB_Keogh_Lowe for pruning power the retrieval of human motion capture data.
- We demonstrate the efficiency and the effectiveness of our method on the various real human motion datasets.

The rest of the paper is organized as follows. In the next section, we review some discussion of related work. In Section 3, we describe the necessary geometric formulas which are utilized in our human motion data mining approach. Section 4 and Section 5 focus on our approach of human motion data mining technique. Section 4 provides an overall view of the proposed human motion retrieval system which includes the chosen human motion features, the technique to measure the similarity between two motions and the low bounding technique to speed up the searching. Section 5 investigates some proposed techniques for more pruning power the retrieval of human motion capture data. Experiments and results discussion are employed in Section 6. We finish with the conclusion and further work in Section 7.

2   Related Work

In computer animation or movie specified effects, data-driven motion synthesis is an important technique to generate the realistic motions from the recorded motion capture data. Due to the large use of motion capture data, the efficient reuse of such data is more and more challenging. In fact, human motion capture data is a kind of high-dimensional time-series and its raw data describe the

spatial information of a large number of markers in the three-dimensional space, thus the problem of human motion data mining is considered as the time-series retrieval. In terms of time-series matching, three main aspects are considered. The first one is how to represent the human motion data. The second one is how to determine the distance function to measure the similarity of two motions and the last one concerns the considered techniques to speed up the searching.

In order to deal with motion capture data, several representations have been proposed. Lee et al. [6] described a two-layer structure for representing human motion capture data, but motion physical features cannot be represented clearly in this case. Chui et al. [7] proposed local spherical coordinates relative to the root orientation as the segments posture of each skeletal segment. But the skeletal segment is represented by two parameters which cannot be used to observe posture of each skeletal segment. In our work, we utilize a spatial invariance representation from the work of Xiao et al. [5]. The representation is based on the bone angle and is effective for human motion retrieval. Opposed to the work of Xiao et al. [5] which considers the human motion as frames, we treat the human motion as a curves cluster of the bone angles, thus the problem of temporal variation could be handled.

In fact, the human motion data mining is a part of time-series data mining in a very large database (VLDB) domain. In terms of the distance function to measure the similarity of the time-series data, there are many proposed methods classified into two main approaches: Model-based approach such as Hidden Markov Model (HMM)[8], Neural Network (NN)[9], and Distance function-based approach, which is differentiated into two sub-approaches, in which the first one is called the Metric functions such as Euclidean distance, Manhattan distance or Normal distance [10] and the second one is called the Non-metric functions such as constrained Dynamic Time Wrapping (cDTW)[14], wLCSS[17], Edit distance with Real Penalty (ERP)[11], Edit Distance on Real sequence (EDR)[12], Spatial Assembling Distance (SpADe)[15], etc. In that domain, the experimental results of Keogh et al., 2002 [14] have to be mentioned. Keogh et al. showed that the performance of the distance function-based approach is much better than that of the model-based approach, for both criteria of accuracy and computational cost. Moreover, the work of Ding et al., 2008 [15] performed a comparison of the major techniques of distance function-based approach by testing their effectiveness on 38 time series data set from a wide variety of application domains. Experimental results obtained show that, in over a dozen of distance measures of similarity among time series data in the literature, cDTW [14][16] is slightly better than the others from accuracy point of view. Moreover, on large data set, computation cost of cDTW is very close to Euclidean distance algorithm. Indeed, cDTW is our chosen distance function to measure the similarity of the human motion capture data.

In terms of speed up the searching, there are two techniques utilized. The first one consists in reducing the length of the human motion by using approximating representations. The other one consists in quickly filtering the negative time-series due to the fact that the number of the positive time-series which

match with the candidature sequence is less much than the number of the negative ones. Many techniques have been proposed in the literature for representing time series with reduced dimensionality. Most of them have been reviewed in work of Ding et al. [15], such as Discrete Fourier Transformation (DFT) [18], Single Value Decomposition (SVD) [18], Discrete Cosine Transformation (DCT) [19], Discrete Wavelet Transformation (DWT) [20], Piecewise Aggregate Approximation (PAA) [22], Adaptive Piecewise Constant Approximation (APCA) [21], Chebyshev polynomials (CHEB) [23], Symbolic Aggregate approXima-tion (SAX) [24], Indexable Piecewise Linear Approximation (IPLA) [25] and etc. In our work, we approach another approximating representation called curve simplification representation which is proposed by Lowe [26]. The experimental result shows the efficiency and effectiveness of that approach in the human motion database [27][28]. Opposed to the original version of the curve simplification algorithm, we propose modified versions adapting with the human motion data mining problem. To speed up the searching by filtering rapidly the negative time-series sequences, lower bounding is popular technique utilized. The main purpose of the using the lower bounding function is due to the fact that, the computational complexity of the measurement distance is in the order of $O(n2)$ in general while using lower bounding technique we are capable of pre-filtering the negative sequences with the complexity in the order of $O(n)$. Thus, we should avoid using the measurement distance for the negative sequences. Moreover, the number of the negative sequences is much more than the number of potential positive ones in the real datasets. Actually, due to the utilized measurement distance of $cDTW$ in the paper, there are several lower bounding functions available for $cDTW$ such as the lower bounding function introduced by Kim et al. [29] or another one proposed by Yi et al. [30]. In addition, Keogh et al. [14] proposed a more efficient lower bounding function referred as LB_Keogh. Their experimental results demonstrate the effectiveness of LB_Keogh in terms of time-series data retrieval. Indeed, our technique is based on the LB_Keogh which is described in Section 4 of the paper.

3   Geometric Formulas for the Human Motion Data Mining Approach Based on Curves Matching

In this section, we describe the necessary geometric formulas which are utilized during the paper. In prior, we describe some definitions relative to our work.

- **Coordinate system of the curve:** The coordinate system of the curve is the 2D space which is defined by vertical axis illustrated in degree unit and horizontal axis illustrated in frame number unit (temporal axis of the human motion).
- **Time-series curve:** In our work the curve of m time-series points is referred as time-series curve which is denoted as follows:

$$C(m) = \{c_i(p_{c,i}, t_{c,i})\}, i = 1..m; t_i > t_j \,\forall i > j \qquad (1)$$

where $t_{c,i}$ is measured in the order of the frame number unit and $p_{c,i}$ is measured in degree unit and m is the length or the dimension of the curve.

Due to the fact that $t_{c,i} > t_{c,j} \forall i > j$, thus the time-series curves are the forward curves or the none-self-intersecting curves. In the paper, we compare curves which have the same frame number value of the beginning point and the end point, that means the $t_{c,1}(t_{c,n})$ of any curve refers the same value. Having defined the necessary terms, we conduct to describe the geometric formulas utilized in our work.

3.1 Area of the Polygon Constituted by Two Time-Series Curves

The first formula in our work concerns the polygon area composed by the two time-series curves U(m) and V(n) to compare. We differentiate two cases. The first case is composed by all configurations of U(m) and V(n) with no intersecting points between them. Then, the polygon is the none-self-intersecting polygon which is illustrated in Figure 1. We denote S_1 as the area of the polygon. Formula 2 is described as below:

$$S_1(U,V) = \left\| \frac{s_1 + s_2 + s_3 + s_4}{2} \right\| \tag{2}$$

where $s_1 = \sum_{i=1}^{m-1} sign(u_i) \times (p_{u,i} \times t_{u,i+1} - p_{u,i+1} \times t_{u,i})$, $s_2 = sign(u_m) \times (p_{u,m} \times t_{v,n} - p_{v,n} \times t_{u,m})$, $s_3 = sign(v_1) \times (p_{u,1} \times t_{v,1} - p_{v,1} \times t_{u,1}$, and $s_4 = \sum_{j=2}^{n} sign(v_j) \times (p_{v,j-1} \times t_{v,j} - p_{v,j} \times t_{v,j-1})$ and $sign(u_i)$ or $sign(v_j)$ is the sign function which obtains the value of 1 or -1 denoted the sign of the polygon edges illustrated in Figure 1 in which, $sign(u_i), \forall i = 1...m-1$ denotes the sign of the edge made by the point ui and u_{i+1}, sign(um) denotes the sign of the edge made by the point um and the point vn, $sign(v_j), \forall j = 2...n$ denotes the sign of the edge made by the point v_{j-1} and the point v_j, and $sign(v_1)$ denotes the sign of the edge made by the point u_1 and the point v_1.

In fact, we obtained the formula as above due to the familiar formula to calculate the area of the none-self-intersecting planar polygon described that: *The area of a parallelogram and triangle can be expressed as the magnitude of the cross-product of two edge vectors* [33]. In the case of non-self-intersecting polygon we could assume that $sign(u_i) = 1$ and the $sign(v_j) = -1$, $\forall i, j$ thus s_1, s_2, s_3, s_4 could be reduced as : $s_1 = \sum_{i=1}^{m-1} (p_{u,i} \times t_{u,i+1} - p_{u,i+1} \times t_{u,i})$, $s_2 = p_{u,m} \times t_{v,n} - p_{v,n} \times t_{u,m}$, $s_3 = p_{v,1} \times t_{u,1} - p_{u,1} \times t_{v,1}$, and $s_4 = \sum_{j=2}^{n} (p_{v,j} \times t_{v,j-1} - p_{v,j-1} \times t_{v,j})$

In practice, a technique with the computational complexity in the order of $O(n)$ can be performed to calculate the area of the none-self-intersecting polygon.

The second case is illustrated in Figure 2, in which two time-series curves may have intersecting points. We refer the polygon of this case as the self-intersecting polygon. Recall the formula 2, we denote S_2 as the area of the self-intersecting polygon thus S_2 is formulated as in Formula 3.

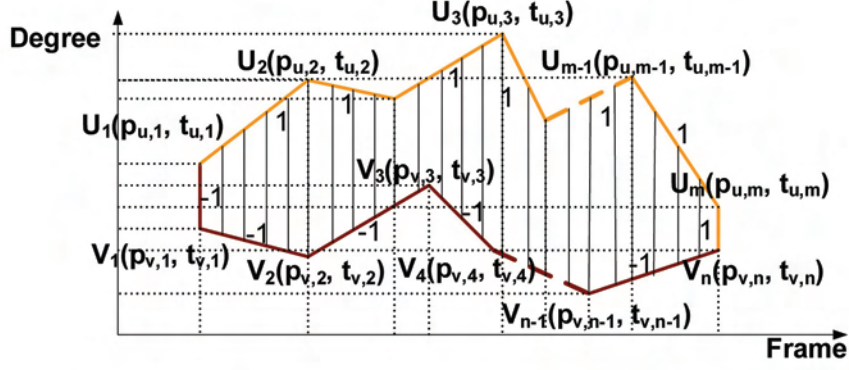$$S_2(U,V) = \left\| \frac{s_1 + s_2 + s_3 + s_4}{2} \right\| \tag{3}$$

Fig. 1. Area of the none-self-intersecting polygon (drawn by the vertical lines) constituted by two time-series curves

where $s_1 = \sum_{i=1}^{m-1} sign(u_i) \times (p_{u,i} \times t_{u,i+1} - p_{u,i+1} \times t_{u,i})$, $s_2 = sign(u_m) \times (p_{u,m} \times t_{v,n} - p_{v,n} \times t_{u,m})$, $s_3 = sign(v_1) \times (p_{u,1} \times t_{v,1} - p_{v,1} \times t_{u,1}$, and $s_4 = \sum_{j=2}^{n} sign(v_j) \times (p_{v,j-1} \times t_{v,j} - p_{v,j} \times t_{v,j-1})$

At first glance, Formula 3 seems to be the same as Formula 2, but in reality the self-intersecting polygon constituted by two curves is added the intersecting points between them, such as the intersecting point $v_2 \equiv u_2$ or the intersecting point $v_4 \equiv u_3$ as illustrated in Figure 2. In this case, the sign function of the edges will be determined based on the sign of the previous edge and the convenience of equalities described in Figure 3. In left visual diagram, the signs of edges (u_{i+1}, u_{i+2}) or (v_{j+1}, v_{j+2}) change due to the signs of their previous edges. In the right one, the signs are not changing.

In practice, intersecting points are unknown when occurring between two curves, so we should have a function to sort out the intersecting points between two edges. Anyways, a technique with the computational complexity in the order of $O(n)$ could be considered to calculate the area of the self-intersecting polygon.

3.2 Area of the Outlier Parts Generated by the Time-Series Curve and the None-Self-Intersecting Polygon Constituted by Two other Time-Series Curves

Having the formulas to calculate the area of both the none-self-intersecting polygon and the self-intersecting polygon, we calculate the area of the outlier parts which are generated by a time-series curve denoted C(l) and a none-self-intersecting polygon constituted by two others time-series curves noted U(m) and V(n). As we described above, the three curves have the same frame number value of the beginning point and the end point. Figure 4 is an example illustrating the formula in which, the outlier parts are drawn by the vertical line
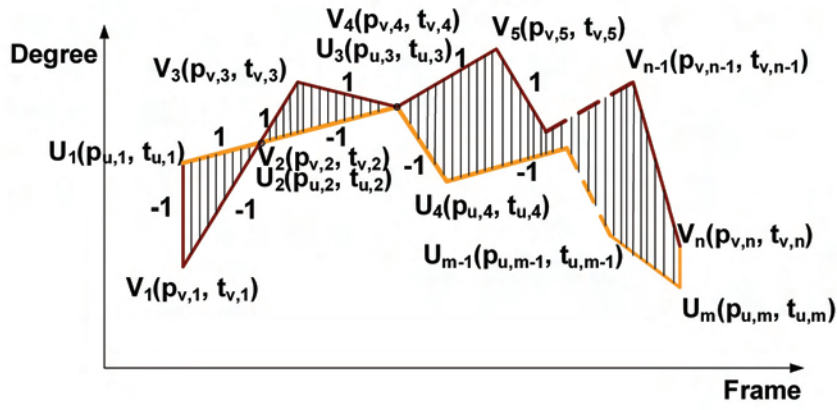
Fig. 2. Area of the self-intersecting polygon (drawn by the vertical lines) constituted by two time-series curves

which consists of three sub-polygons s_2, s_5 and s_7. S(C,U,V) is the area of the outlier parts. S1(U, V) is the area of the none-self-intersecting polygon constituted by U(m) and V(n). S2(U, C) and S2(C, V) are noted the areas of the self-intersecting polygon constituted by U(m) and C(l), and the self-intersecting polygon constituted by C(l) and V(n), respectively. The formula to calculate S(C,U,V) is described as below in Formula 4.

$$S(C, U, V) = \frac{S_2(U, C) + S_2(C, V) - S_1(U, V)}{2} \tag{4}$$

We have described the necessary geometric formulas which are utilized in the paper. In the next sections, we present our approach to retrieve a human motion data based on the curves matching.

4 Human Motion Data Mining Based on the Curves Matching Approach

As we describe above, there are three main aspects concerned to the retrieval of the human motion data included the representation of the human motion data, the chosen distance function to measure the similarity and the considered techniques to speed up the searching. In Section 4 and Section 5, we investigate to describe our approach for these three aspects.

4.1 Spatial-Temporal Invariance Representation for Human Motion Capture Data

A simplified kinematic chain model is defined as Figure 5, which contains 14 joints that are constructed as a tree diagram. Eleven bones are extracted as the
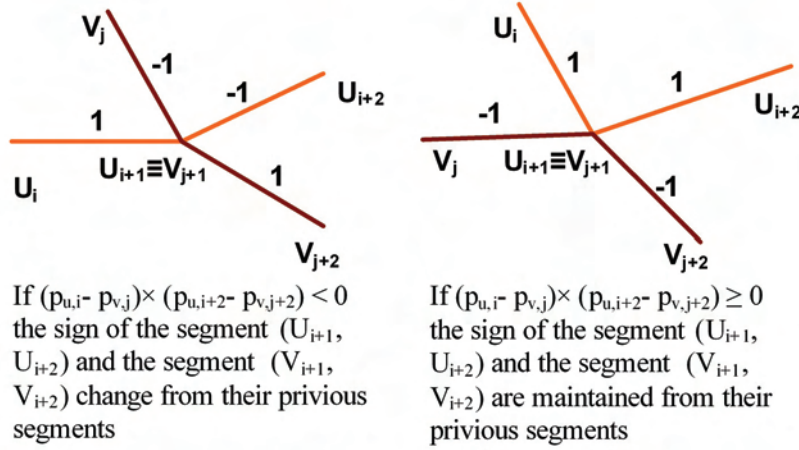
Fig. 3. Two considered cases to determine the changing of the sign functions of the edge from the its previous edge

objects to represent motion feature, including 10 bones in the human limbs and a central bone connected by *root* and *chest* joints as a reference bone. Each bone is defined as a vector from the upper joint to the lower joint in the human skeleton. For every limb bone, the bone angle is defined as the angle between the limb bone and the central bone.

$$\theta_i^k = \cos^{-1}(\frac{\boldsymbol{B}_i^{(k)}.\boldsymbol{B}_i^{(Center)}}{\left|\boldsymbol{B}_i^{(k)}\right| \times \left|\boldsymbol{B}_i^{(Center)}\right|}), k = 1..10 \tag{5}$$

where $\boldsymbol{B}_i^{(Center)}$ represents the central bone at the ith frame and θ is in the interval $[0, \pi]$. Consequently, by using the bone angle, we obtain a spatial invariance representation for human motion data. In addition, from Formula 5 the motion hereafter could be treat as a cluster of ten curves, in which each curve describes a bone motion. Hence, we can introduce the human motion M data as follows:

$$M = \left\{C^k(\theta_1^k, \theta_2^k, ...\theta_{m_k}^k)\right\}, k = 1..10 \tag{6}$$

In fact, the curves C_k are processed independently to carry out the technique of lower bounding or the technique of the dimensional simplification. Therefore, the curves C_k are considered as the temporal invariance representation of the human motion data. With such representation, our approach is a spatial-temporal invariance method. In practice, instead of using the motion M hereafter we utilize the curve C to describe our approach more easily. More, the similarity of the human motions could be treated as the similarity of the curves or the curves matching problem.
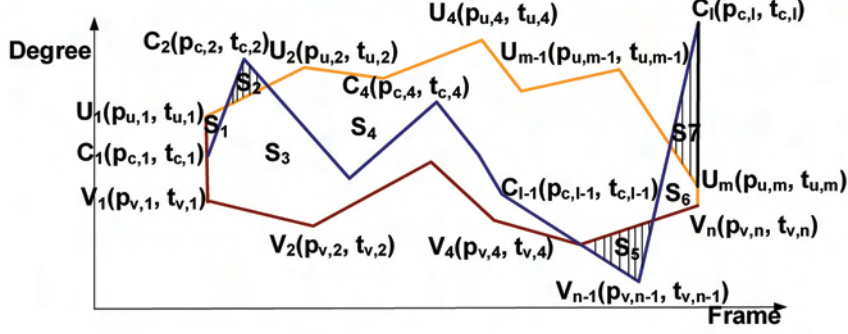
Fig. 4. Area of the outlier parts between a curve C(l) and a none-self-intersecting polygon constituted by two other curves U(m) and V(n). The outlier parts are drawn by the vertical lines. Denote S(C,U,V) as the area of those parts, si is the area of the ith sub-polygon generated by the curves illustrated as in Figure 4. We have:$S(C, U, V) = s_2 + s_5 + s_7$; $S_1(U, V) = s_1 + s_3 + s_4 + s_6$; $S_2(U, C) = s_1 + s_2 + s_4 + s_5 + s_7$; $S_2(C, V) = s_3 + s_2 + s_5 + s_6 + s_7$ Hence, we can deduce that: $S(C, U, V) = \frac{S_2(U,C)+S_2(C,V)-S_1(U,V)}{2}$

4.2 Distance Function to Measure the Similarity between Two Human Motions: Constrained Dynamic Time Warping Technique

Our chosen distance function to measure the similarity of the human motion is based on the constraint Dynamic Time Warping ($cDTW$) technique which is introduced in many time-series data matching references [14][16]. The $cDTW$ technique can be described as follows. Given two time-series curves due to the formula 1 $U(m) = (u_1, u_2, ..., u_m)$and $V(n) = (v_1, v_2, ..., v_n)$ and the time warping constraint ε, the constrained Dynamic Time Warping $cDTW$ is defined recursively as follows [16]:

$$Dist_r(u_i, v_j) = \begin{cases} D_{base}(u_i, v_j) \, if \, |i - j| \leq r \\ \infty \qquad otherwise \end{cases}$$
$$cDTW(\phi, \phi, r) = 0$$
$$cDTW(U, \phi, r) = cDTW(\phi, V, r) = \infty$$

$$cDTW(U, V, r) = Dist_r(First(U), First(V)) + \min \begin{cases} cDTW(U, Rest(V), r) \\ cDTW(Rest(U), V, r) \\ cDTW(Rest(U), Rest(V), r) \end{cases}$$
$$(7)$$

where $r = \varepsilon \times t_{u,m} = \varepsilon \times t_{v,n}$ and ϕ is the empty curve,$First(U) = u_1$, $Rest(U) = u_2, u_3, ...u_m$ and D_{base} denotes the distance between two vertices of the curve. In our work, we utilize the Manhattan distance which is a special case of the Lm distance with index value α is equal to 1 defined as follows:

$$L_{\alpha=1} \left(u_i(p_{u,i}, t_{u,i}), \, v_j(p_{v,j}, t_{v,j}) \right) = \left(|p_{u,i} - p_{v,j}|^{\alpha=1} + |t_{u,i} - t_{v,j}|^{\alpha=1} \right)^{\frac{1}{\alpha}=1} \quad (8)$$
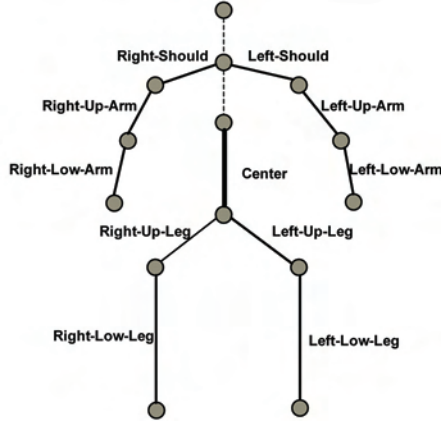
Fig. 5. Human skeleton and segments separated

4.3 Lower Bounding Technique for Speeding Up the Human Motion Data Retrieval

As we describe previously, the main purpose of the using the lower bounding technique is to filter more quickly the negative time-series curves. Our lower bounding function is based on the LB_Keogh described as follows. Given a time-series curve as the formula 1, $C(m) = \{c_i(p_{c,i}, t_{c,i})\}, i = 1..m; t_i > t_j \forall i > j$ and a local constraint time warpingε, we note that $r = \varepsilon \times t_{c,m}$ and we use the term r to define two new time-series curves, U(m) and L(m):

$$
\begin{aligned}
&U(m) = \{u_i(p_{u,i}, t_{u,i})\} \; and \; L(m) = \{l_i(p_{l,i}, t_{l,i})\} \\
&where \, t_{u,i} = t_{l,i} = t_{c,i} \\
&p_{u,i} = \max(p_{c,i-r} : p_{c,i+r}) \\
&p_{l,i} = \min(p_{c,i-r} : p_{c,i+r})
\end{aligned}
\tag{9}
$$

U(m) and L(m) stand for Upper and Lower, respectively. Figure 6 illustrates the created U and L with ε=20% of the length of the curve C. As we can see in the figure, U and L form a bounding envelope curves to enclose C from above to below. An important property of the bounding curves U and L is the following:

$$
\forall i, p_{u,i} > p_{c,i} > p_{l,i}
\tag{10}
$$

Having defined the bounding curves U and L of the curve C and the Manhattan distance defined in Formula 8, we now use them to define a lower bounding measure for cDTW. Given two time-series curves C(m) and V(n), the bounding measure of V and the bounding envelopes of C are defined as follows:

$$
LB\_Keogh(C, V) = \sum_{i=1}^{t_{u,m}} \begin{cases} |p_{v,i} - p_{u,i}| \; if \, p_{v,i} > p_{u,i} \\ |p_{v,i} - p_{l,i}| \; if \, p_{v,i} < p_{l,i} \\ \quad 0 \quad\; otherwise \end{cases}
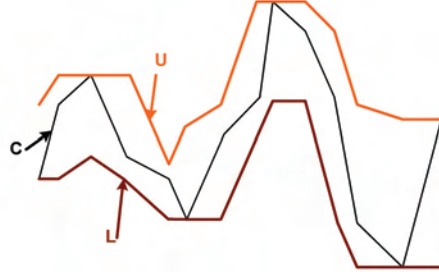\tag{11}
$$

Fig. 6. An illustration of the time-series curves U and L, created for the time-series curve C(m) by using the LB_Keogh lower bounding function in which, r is a constant value due the length of $C(m) : r = 0.2 \times m$

This function can be visualized as the Manhattan distance between any parts of the query matching curve not falling within the envelope and the nearest (orthogonal) corresponding section of the envelope. Thus, the function is the area of the outlier parts generated by the curve V and the none-self-intersecting polygon constituted by two curves U and L, illustrated in Figure 7. Hence, we recall Formula 4 to calculate the LB_Keogh(C,V) in practice:

$$LB\_Keogh(C, V) = S(V, U, L) = \frac{S_2(U, V) + S_2(V, L) - S_1(U, L)}{2} \qquad (12)$$

Now we will explain how utilize the LB_Keogh function to filter the negative curves to speed up the searching. In prior, we recall the work of Keogh et al. [16] which proved that:

$$LB\_Keogh(C, V) \leq cDTW(C, V) \qquad (13)$$

Given a curve C and a threshold δ, we define a curves V as the positive matching curve with C if the inequality as follows is convenient:

$$cDTW(C, V) \leq \delta \qquad (14)$$

Hence, we can see that the curve V will be the negative curve if

$$LB\_Keogh(C, V) > \delta \qquad (15)$$

. In general, that inequality is utilized to filter the negative curves but our work will be continued for speed up the human motion data mining.

In this section, we have described an overall view of the human motion data mining in which, we conducted the human motion data mining as the curves matching problem by using the spatial-temporal representation for the human motion data. We also have presented the cDTW to measure the similarity of the curves and the technique to speed up the searching based on the lower bounding function called LB_Keogh for filtering rapidly the negative curves. In the next section, we describe another technique to speed up the searching based on the dimensional reduction technique called the curve simplification.
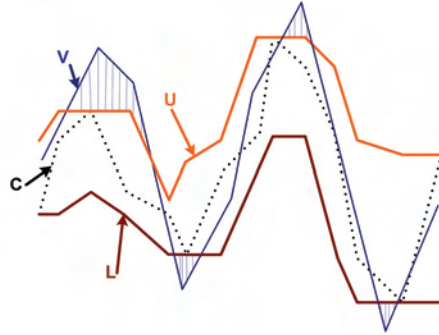
Fig. 7. An illustration of the lower bounding function LB_Keogh between the original curve C (shown dotted) and the query curve V calculated by the area of the outlier parts generated by the curve V and the none-self-intersecting polygon constituted by the bounding envelope U and L

5 Techniques of Dimensionality Reduction Based on the Curve Simplification

As we describe above, the technique to reduce the dimension of the curves is utilized to speed up the searching besides the technique of using the lower bounding. In the approach of the curves matching problem, we utilize the curve simplification technique to reduce the dimension. The technique is proposed by Lowe [26] and is utilized efficiently to the key-frame extraction of the human motion data [1][5][27]. Due to the fact that, in our work, two axis of the coordinate system for the time-series curves do not have the same unit, while vertical axis is represented in the degree unit the horizontal one is represented in the frame number unit. Therefore instead of using the Euclidean distance of the point to a line as in the original version of Lowe technique, we utilize a distance measure of the length of the segment defined by the given point and the point lied in the line which has the same frame number value (orthogonal point). The idea is illustrated in Figure 8.

5.1 Lowe Algorithm to Simplify the Candidate Curve

Step 1: Set the first and the last points as key points, creating 2 key points.

Step 2: Find the highest absolute distance point, which would have the distance called the error distance. If the error distance is lower than a specified thresholdσ, stop further subdividing this interval. Otherwise, create a new key point at the point with the highest error distance.

Step 3: Sub-divide the current state into two smaller segments, a segment between the beginning key point and the newly created middle key point, and another segment between the middle key point and the ending key point. Assume that the new segments now have only 2 key points and restart the algorithm from step 2 for both segments.
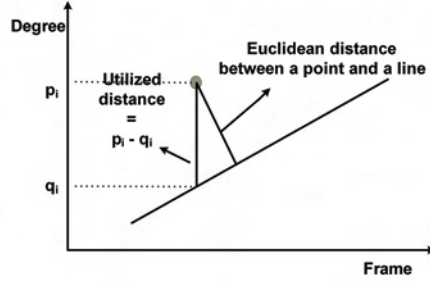
Fig. 8. An illustration of the utilized distance instead of using the Euclidean distance as in the original version of the Lowe technique

Figure 9 shows the modified algorithm to simplify the curve step by step. In practice, that modified algorithm is applied to the original candidate curves. Given a time-series curve C(m) with m is the dimension of the curve, we note $\bar{C}(\bar{m})$ is the simplified curve of C with the dimension of $\bar{m}$. In general, we always maintain the important property of the any time-series curve C(m) which is the tc,1 or tc,m referring to the same value.

In the next part, we will describe another modified Lowe algorithm to simplify the curve which adapts the bounding property of the envelope curves.
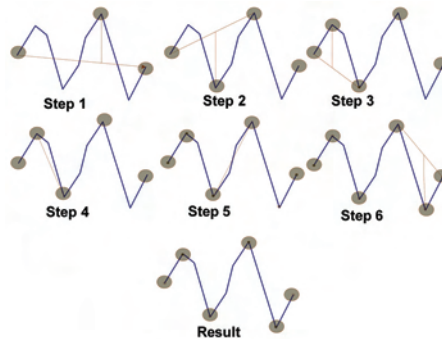

Fig. 9. An illustration of the modified curve simplification in which, the chosen points are highlighted

5.2 Modified Lowe Algorithm to Simplify the Bounding Envelope Curves

Opposed to the modified algorithm to simplify the When we simplify the bounding curves we have to ensure that the original curve is always enclosed by the

bounding curves. Therefore, we proposed another modified Lowe algorithm for bounding curves simplification. We recall the utilized distance illustrated in Figure 10 in which, we occupy the value with the sign of the subtraction $p_i - q_j$. We differentiate into two cases, one for the upper bounding curve U and another for the lower bounding curve L.

Modified simplification technique for upper bounding curve is described as follows:

Step 1: *Set the first and the last points as key point, creating 2 key points.*

Step 2: *Find the highest positive distance point. If the point is found, create a new key point at the point with the highest positive distance; otherwise find the lowest negative distance point which would have the distance called the error distance. If the absolute value of the error distance is lower than a specified thresholdσ, stop further subdividing this interval. Otherwise, create a new key point at the point with the lowest negative error distance.*

Step 3: *Sub-divide the current state into two smaller segments, a segment between the beginning key point and the newly created middle key point, and another segment between the middle key point and the ending key point. Assume that the new segments now have only 2 key points and restart the algorithm from step 2 for both segments.*

Modified simplification technique for lower bounding curve is described as follows:

Step 1: *Set the first and the last points as key point, creating 2 key points.*

Step 2: *Find the lowest negative distance point. If the point is found, create a new key point at the point with the lowest negative distance; otherwise find the highest positive distance point which would have the distance called the error distance. If the error distance is lower than a specified thresholdσ, stop further subdividing this interval. Otherwise, create a new key point at the point with the highest positive error distance.*

Step 3: *Sub-divide the current state into two smaller segments, a segment between the beginning key point and the newly created middle key point, and another segment between the middle key point and the ending key point. Assume that the new segments now have only 2 key points and restart the algorithm from step 2 for both segments.*

5.3   Pruning Power of the Human Motion Capture Data Mining

Having defined the dimensionality reduction of the candidate curves and the bounding envelope curves of the query curves, now we explain how to utilize them for pruning power of the human motion capture data mining. We recall Formula 12 to calculate the LB_Keogh of the query curve C and the candidate curve V in which, $LB\_Keogh(C, V) = S(V, U, L)$is the area of the outlier parts generated by the curve V and the non-self-intersecting polygon constituted by two bounding curves of C. We occupy the simplified curve $\bar{V}$, the simplified curves $\hat{U}$ and $\hat{L}$to formulate a new bounding measure denoted LB_Keogh_Lowe(C,V) which is
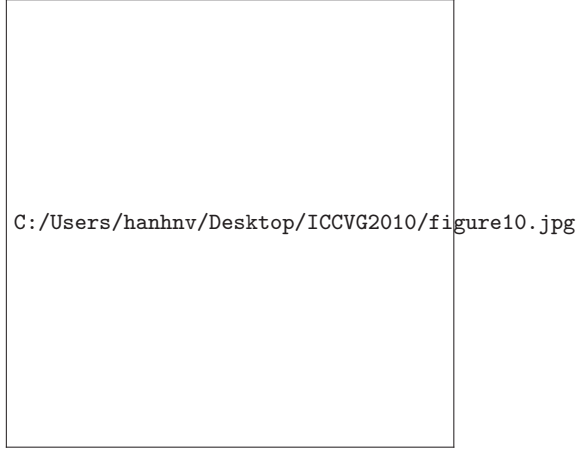
Fig. 10. Illustrates an example of the carrying out the modified curve simplification technique to the bounding curves, in which we note the simplified curve of the upper bounding as $\hat{U}$ and that of the lower bounding as $\hat{L}$

describe by the formula:

$$LB\_Keogh\_Lowe(C,V) = S(\bar{V},\hat{U},\hat{L}) = \frac{S_2(\hat{U},\bar{V}) + S_2(\bar{V},\hat{L}) - S_1(\hat{U},\hat{L})}{2} \quad (16)$$

The difference between LB_Keogh and LB_Keogh_Lowe is illustrated in Figure 11. In practice, the computation cost of LB_Keogh_Lowe which performs the curve simplification on both of the bounding curves and the query curve is lower than that of the LB_Keogh.

To filter rapidly the negative curves, given a threshold δ, instead of using the inequality 15 we utilize another inequality due to the LB_Keogh_Lowe in which, the query curve V will be a negative curve if

$$LB\_Keogh\_Lowe(C,V) > \delta \quad (17)$$

In practice, given a dataset of the human motions, we perform a pre-processing to obtain the dataset of query curves C, the lower bounding curves U and V, the reduced dimensionality lower bounding curves $\hat{U}$ and $\hat{V}$. That pre-processing is considered as the offline working. Hence, it does not impact the performance of the system. More, we prefer using the δ as the function of the $t_{c,m}$ which means $\delta := \delta \times t_{c,m}$ due to the convenience of choosing δ

We summarize this section by Figure 12 illustrating the proposed algorithm to handle the human motion data best match retrieval based on the curves matching.

In next section, we describe our experimental evaluation of our proposed approach with various real human motion datasets.

6 Experiments and Results

This section describes the experiments carried out to analyze the effectiveness of the curve simplification technique in term of the dimensionality reduction and the proposed lower bounding function which is based on that technique. In addition, we perform an experiment to verify the efficiency of the human motion data mining. The experiments were executed on an Intel Pentium Dual-Core 2.1GHz Window PC with 2GB RAM and the program is written in C++ Language.

To analyze the performance of the dimensionality reduction technique based on the curve simplification, we define a pruning scale P_1 which is described as follows:

$$P_1 = \frac{Number\ of\ the\ points\ of\ the\ reduced\ curves}{Number\ of\ the\ point\ of\ the\ original\ curves\ in\ database} \times 100(\%) \quad (18)$$

To analyze the effectiveness of the proposed lower bounding function based on the curve simplification technique, we define another pruning power scale P_2 described as follows:

$$P_2 = \frac{Number\ of\ the\ human\ motions\ that\ do\ not\ require\ full\ wDTW}{Number\ of\ the\ human\ motions\ in\ databased} \times 100(\%)$$
$$(19)$$

6.1 Data Preparing

We gathered more than 500Mb of the human motion dataset from Mocap-Club.com [31] with various kinds of human motion activities such as the walking, running, dancing, kung-fu and etc which are recorded in the frequency of 60Hz with more than 42 markers attached on the human body.Due to the proposed approach of the human motion representation, we extracted from the original dataset the 3D position of 14 joints of the kinematic chain model as described in Figure 5. Thus, we transformed them to the bone angle space and sort them so that each human motion is described by 10 curves of the bone angle. The obtained dataset differed significantly in size and length. In order to produce meaningful results, we reorganized obtained dataset in the bone space. We derived six sets of data, each containing 270 human motion sequences, with variable lengths of 32, 64, 128, 256, 512 and 1024, respectively. Short sequences were gathered by using a scale factor of the length of original data and the desired lengths of extracted data while long sequences were produced by concatenating original sequences. In practice, all experiments were conducted on these derived dataset.

6.2 Experiments and Results

Performance of the dimensionality reduction technique based on the curve simplification. Figure 13(a, b, c) illustrate the performance of the curve simplification P1 due to the length of the human motion. The Figure 13a is the

outcome when we carry out the technique for the original dataset while Figure 13b and Figure 13c are the results carried out for the lower bounding curves. As shown in the figure, the performance of the curves simplification technique increased with the length of the data suggesting. In the lower bounding curves, the differences performances among the specified thresholds are not considerable thus we should choose the specified threshold as small as possible. In practice, to analyze the pruning power performance and the query time, we choose the threshold for lower bounding curves as $\sigma = 5^0$. The specified threshold to reduce the dimension of the original candidate data is chosen due to the threshold δ of the LB_Keogh_Lowe. We recall the work of Baek et al. [32] which developed experiment to choose the threshold for the human joint of the kinematic chain model. From this work, the suggested error interval for human joint is (100,200) thus we choose the specified threshold $\sigma = 10^0$ which is illustrated as the green curve in Figure 13a. Moreover, we also choose the threshold $\delta$of the LB_Keogh_Lowe 100 due to the reference work of Baek.

Pruning power performance. Having defined the threshold σ and δ, now we can verify the pruning power performance of the LB_Keogh_Lowe bounding function due to the capability of filtering the negative human motions. In fact, we perform the 1-nearest neighbor search using the sequential scan technique. A random human motion was chosen from the data set to act as the query and the remaining 269 human motions acted as the data. The search carried out 50 trials on each different lengths of the human motion. We recall Formula 20 to measure the pruning power for query human motion. The average of the 50 queries was reported as the pruning power of each different lengths of the dataset. Figure 14 shows how the pruning power averaged of the proposed LB_Keogh_Lowe. Lower bounding function varies as the lengths of the data in the human motion dataset in which, 92% of the human motion of length 1024 and 67% of the human motion of length 32 did not require computation of the actual time warping distances. The promising pruning power greatly reduces the querying time. We conducted experiments to measure the time required for the query evaluation of the human motion in different lengths. In prior, we should recall the work of Keogh et al. [16] to confirm the time warping constraintε. Due to the experimental result of Keogh the threshold $\varepsilon$is the most efficient with the value of 20% of the length of the real time-series data. Thus we denote ε=0.2.

Query time analysis. Having defined of $\sigma, \delta, \varepsilon$, we carried out the experiment to analyze the query time of the human motion. Similar to the pruning power analysis, the average value of 50 trials of each different lengths of the human motion are reported. Figure 15 illustrates the compared results of query time analysis in the human motion retrieval between the lower bounding approach and none-lower bounding approach. In fact, the average query time is consistently reduced and the difference between two query time curves is considerable, in which to search the 1-nearest, the human motion of length 256, our approach just needs 2.99 seconds averaged while the none-lower bounding approach needs 4,84

seconds averaged, thus we achieved 38% more efficient in time cost processing. To search the longest human motion of length 1024, our approach needs 20,92 second averaged while the none-lower bounding approach needs 64,96 seconds averaged. Hence, we also achieved 68% more efficient in time cost processing.

7   Conclusion and Future Work

In our paper, we have proposed a novel approach to retrieve a human motion data. The approach is based on the computational geometry and conducted the human motion data mining to the curves matching problem. We have introduced a spatial-temporal invariance representation for the human motion data. We have described the modified curve simplification algorithms adapted to the purpose of the human motion data mining and proposed the use of a lower bounding technique noted LB_Keogh_Lowe to speed up the searching. We have demonstrated the efficiency and effectiveness of our method on various real human motion datasets. Due to the obtained results, our approach can be applied in others contexts of the time-series data retrieval. For future work, we would like to carry out our approach to various kinds of time-series data to confirm the efficiency and the effectiveness and to compare with the other approaches in terms of the time-series data approximating representations and lower bounding distances.

8   Acknowledgments

References

1. Muller, M., Roder, T., Clausen, M.: Efficient Indexing And Retrieval of Motion Capture Data Based on Adaptive Segmentation. Proceedings of the 4th Intl. Workshop on Content-Based Multimedia Indexing, Riga, Latvia, (2005)
2. Giese, M. A., Poggio, T.: Morphable models for the analysis and synthesis of complex motion pattern. International Journal of Computer Vision, Volume 38, Number 1, 59–73, (2000).
3. Kovar, Lucas, Gleicher, Michael.: Automated extraction and parameterization of motions in large data sets. Proceedings of ACM SIGGRAPH, (2004).
4. Keogh, E., Palpanas, T., Zordan, V. B., Gunopulos, D., Cardle, M.: Indexing large human-motion databases. Proceedings of the 30th VLDB Conference, Toronto, Canada, 780–791, (2004).
5. Jun Xiao, Yueting Zhuang, Tao Yang, Fei Wu.: An efficient keyframe extraction from motion capture data. In book chapter of Advances in computer graphics, 494–501,(2006).

6. Jehee Lee, Jinxiang Chai, Paul, S., Reitsma, A., Jessica, K.: Hodgins and Nancy S. Pollard. Interactive Control of Avatars Animated with Human Motion Data. In proceedings SIGGRAPH, San Antonio, Texas, 491–500 (2002).
7. Chui, C. Y., Chao, S. P., Wu, M., Y., Yang, S. N., Lin, C.: Content-based Retrieval for Human Motion Data. Journal of Visual Communication and Image Representation, 16(3), 446–476 (2003).
8. Croitoru, Peggy Agouris, Anthony Stefanidis.: 3D trajectory matching by pose normalization. Geographic Information Systems Proceedings of the 13th annual ACM international workshop on Geographic information systems Bremen, Germany, (2005).
9. F Porikli.: Trajectory Distance Metric Using Hidden Markov Model based Representation. In proceeding of IEEE European conference on Computer Vision, (2004).
10. Berndt, J., Clifford, J.: Using dynamic time warping to find patterns in time series. In KDD Workshop, (1994).
11. Chen, L., Ng, R. T.: On the marriage of lp-norms and edit distance. In VLDB, (2004).
12. Chen, L., Ozsu, M. T., Oria, V.: Robust and fast similarity search for moving object trajectories. In SIGMOD Conference, (2005).
13. Chen, Y., Nascimento, M. A., Ooi, B. C., Tung, A. K. H.: SpADe: On Shape-based Pattern Detection in Streaming Time Series. In ICDE, (2007).
14. Keogh, E.: Exact Indexing of Dynamic Time Warping. In proceeding of VLDB Conference, (2007).
15. Ding, Trajcevski, H., Scheuermann, G., Wang, P., Keogh, E.: Querying and Mining of Time Series Data Experimental Comparison of Representations and Distance Measures. In Proc of the 34th VLDB, 1542–1552 (2008).
16. Keogh, E., Ratanamahatana, C. A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst.,7(3), (2005).
17. Keogh, E., Kasetty, S.: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Min. Knowl. Discov., 7(4), (2003).
18. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast Subsequence Matching in Time-Series databases. In SIGMOD Conference, (1994).
19. Chan, K. P., Fu, A. W-C.: Efficient Time Series Matching by Wavelets. In ICDE, (1999).
20. Keogh, E., Chakrabarti, K., Pazzani, M. J., Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowl. Inf. Syst., 3(3), (2001).
21. Keogh, E., Chakrabarti, K., Mehrotra, S., Pazzani, M. J.: Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. In SIGMOD Conference, (2001).
22. Keogh, E., Chakrabarti, K., Pazzani, M. J., Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowl. Inf. Syst., 3(3), (2001).
23. Cai, Y., Ng, R. T.: Indexing spatio-temporal trajectories with chebyshev polynomials. In SIGMOD Conference, (2004).
24. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Discov., 15(2), (2007).
25. Chen, Q., Chen, L., Lian, X., Liu, Y., Yu, J. X.: Indexable PLA for Efficient Similarity Search. In VLDB, (2007).
26. Lowe, G.: Three-dimensional object recognition from single two dimensional images. Artificial Intelligence, vol. 31, no. 3, 355–395, (1987).

27. Lim, L. S., Thalmann, D.: Key-posture extraction out of human motion data by curve simplification. Proceedings of the 23rd Annual EMBS International Conference, Istanbul, Turkey, (2001).
28. Onder, O., Gudukbay, U., Ozguc, B., Erdem, T., Ozkan, C.: Keyframe Reduction Techniques for Motion Capture Data. In proceeding of 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, (2008).
29. Kim, S., Park, S., Chu, W.: An Index-based approach for similarity search supporting time warping in large sequence databases. In Proc 17th International Conference on Data Engineering,607–614 (2001).
30. Yi, B., Jagadish, K., Faloutsos, H.: Efficient retrieval of similar time sequences under time warping. In ICDE, 23–27, (1998).
31. http://www.mocapclub.com/Pages/Library.htm
32. Baek,S., Lee, S., Kim, G. J.: Motion retargeting and evaluation for VR-based training of free motions. In journal of visual computer, Volume 19, Issue 4, 222–224, (2003).
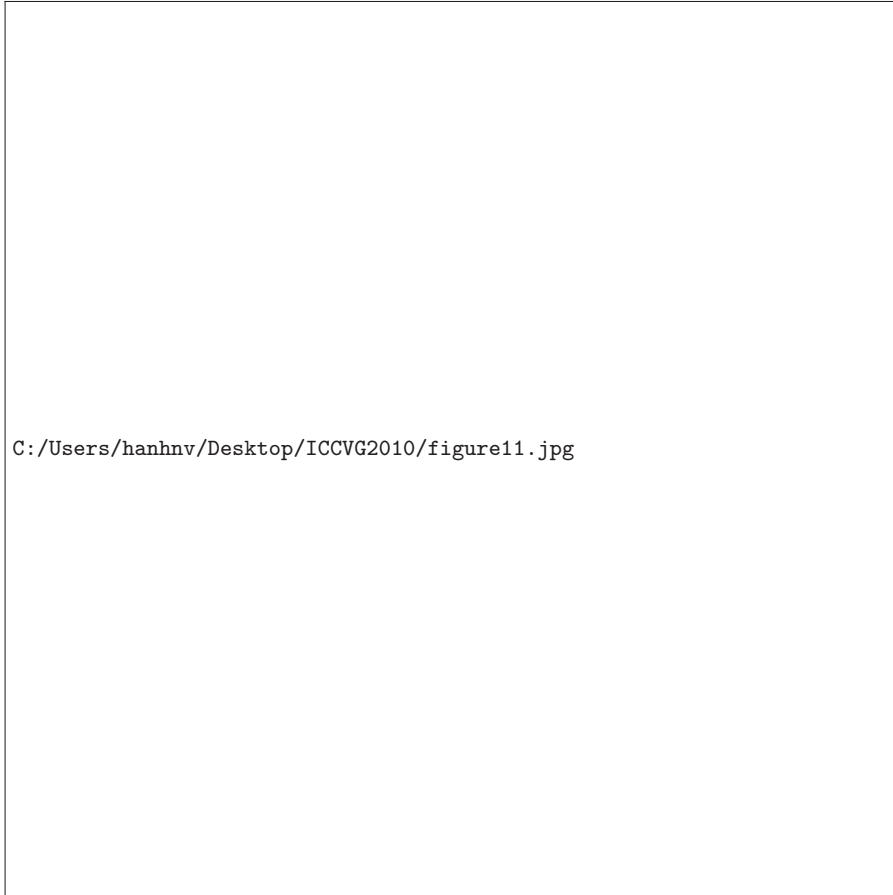33. http://softsurfer.com/Archive/algorithm_0101/algorithm_0101.htm

C:/Users/hanhnv/Desktop/ICCVG2010/figure11.jpg

Fig. 11. Comparison of LB_Keogh lower bounding distance (upper visual diagram): $LB\_Keogh(C,V) = S(V,U,L) = \frac{S_2(U,V)+S_2(V,L)-S_1(U,L)}{2}$ and LB_Keogh_Lowe lower bounding distance (lower visual diagram):$LB\_Keogh\_Lowe(C,V) = S(\bar{V},\hat{U},\hat{L}) = \frac{S_2(\hat{U},\bar{V})+S_2(\bar{V},\hat{L})-S_1(\hat{U},\hat{L})}{2}$

C:/Users/hanhnv/Desktop/ICCVG2010/figure12.jpg

Fig. 12. An illustration of the algorithm to best match retrieval of the human motion data based on the curves matching
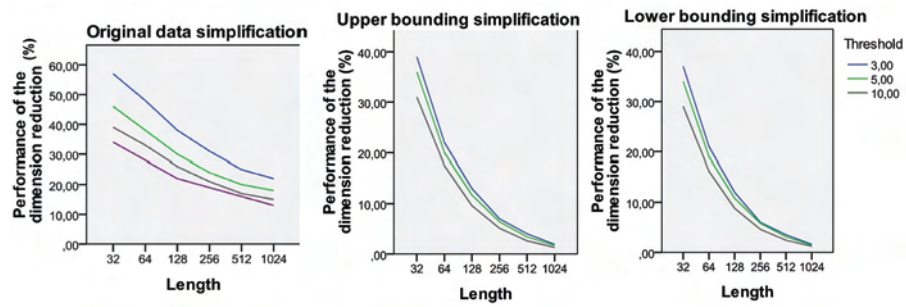

Fig. 13. (a, b, c) An illustration of the performance of the dimensionality reduction technique to the real human motion dataset. Fig 13a is outcome carried out the original human motion; Fig 13b, 13c are results carried out the upper bounding and lower bounding curves, respectively. The specified threshold is in degree.
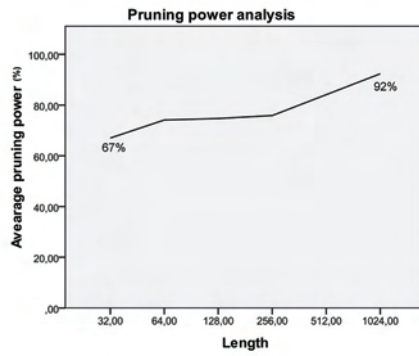
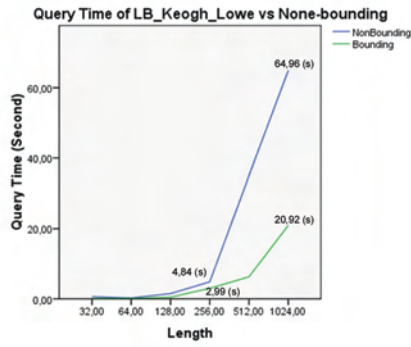**Fig. 14.** Average pruning power in the human motion dataset

Fig. 15. Comparison the average query time between LB_Keogh_Lowe bounding and none bounding technique