Entropy-Based Motion Extraction for Motion Capture Animation

Clifford K. F. So and George Baciu

Department of Computing - The Hong Kong Polytechnic University

Email: {cskfso, csgeroge}@comp.polyu.edu.hk

Abstract

In this paper we present a new segmentation solution for extracting motion patterns from motion capture data by searching for critical keyposes in the motion sequence. A rank is established for critical keyposes that identifies the significance of the directional change in motion data. The method is based on entropy metrics, specifically the mutual information measure. Displacement histograms between frames are evaluated and the mutual information metric is employed in order to calculate the inter-frame dependency. The most significant keypose identifies the largest directional change in the motion data. This will have the lowest mutual information level from all the candidate keyposes. Less significant keyposes are then listed with higher mutual information levels. The results show that the method has higher sensitivity in the directional change than methods based on the magnitude of the velocity alone. This method is intended to provide a summary of a motion clip by ranked keyposes, which is highly useful in motion browsing and motion retrieve database system.

Keywords: motion capture, animation, entropy, mutual information, motion database.

Introduction

The advent of motion capture systems has had a significant impact on the speedy generation of motion for 3D characters for animation and clinical studies in gate analysis. Current motion capture devices capture human motion at high frequency [1] and the motion can be retargeted to computer characters for realistic movements. Applications of motion capture are continuously growing in number and range from crowd generation and controlling the motion of synthetic characters in large scale movie productions to short TV commercials, 3D computer games, bio-kinematics analysis in sports and clinical studies, simulations and preservation of style in specific motions that can be generated by unique individuals, such as for example variations of Kong-Fu in martial arts, dancing and acrobatics. However, motion generation often involves long motion capture sessions and careful editing in order to adapt and fulfill a particular behavior of a character. The captured motion is often archived and annotated for future use. The motion database is probably the most important asset in a production studio. Recorded motions are frequently recalled from the database and new motions are reconstructed from the given clips.

The work of this paper is motivated by the need to work with larger motion databases that require frequent motion storage and retrieval operations. The fundamental questions that we attempt to answer are:

How can we efficiently store all the captured motion into a motion database?

What is the most meaningful way to summarize a motion clip other than text annotation?

How can we retrieve motion segments in a more intuitive way than simply using text annotation?

What features should be extracted from the motion clips for an efficient indexing strategy?

We attempt to address these questions by proposing a new method that first identifies significant changes in the motion patterns and establishing a entropy based ranking in order to extract meaningful motion segments. We define *a keypose* to represent a time sample of the 3D motion such that the motion that occurs prior to the keypose is significantly different from the motion that occurs after the keypose. A keypose gives a visual summary of the motion and identifies a unique posture during the change between two continuous motion changes. A keypose contains more information than text annotation. The use of keyposes is similar to a "keyframe" in the traditional animation, however a keypose is different from a keyframe in that it is the representation of an "extreme" pose at the time when the motion has to pass through from one continuous pattern to another and, therefore, it has a substantial directional change. Thus, a keypose serves as the signature of the motion like the poses featured in dancing motion, kung-fu motion and simple walking or running cycles.

We first solve the keypose extraction problem by using a mutual information method to identify critical keyposes from a given motion sequence. The result of this extraction is a list of ranked critical keyposes that have been detected in a large motion segment. Keyposes determine a substantial directional change in the motion pattern. We use an entropy measure, specifically the mutual information measure, to quantify the amount of information that is passed from one frame to another. A large difference in the directional movement between two frames leads to a weak inter-frame dependency and hence a lower level of mutual information. Our algorithm ranks the keyposes based on their corresponding mutual information levels. These levels are related to the significance in the directional change of the motion. The rank of the keyposes can then be used further in the indexing of the motion retrieval system.

We organize the paper as follows: We first present some background and related work. We give the entropy metric and the details of our work in the next section. We will present the result and compare to the previous work, and gives other possible applications. The final section concludes the paper and suggests future directions.

Related work

Recently, Li et al. [2] presented a motion retrieval system based on a hierarchical motion index tree. The similarity between the sample and the motion in the sub-library is calculated through an elastic match. In order to improve the efficiency of the similarity calculation, an adaptive clustering-based keypose extraction algorithm is adopted. The keyposes are used in the calculation instead of the whole motion sequence. Our work differs from Li et al by searching and identifying the "extreme" poses rather than the most similar poses at the centroids.

Some recent research work also suggests the concept of a motion database. These techniques are mainly motion re-synthesis techniques that automatically generate new motions from existing motion clips. Some algorithms re-synthesize a new motion based on non-annotated motion database with user-specified constrains [3-8]. These algorithms synthesize a new motion that follows a path, go to a particular position, or perform certain activity at a specified time. The motion databases referred to, however, are limited to a set of motion samples and targeted to specific motion re-synthesis needs. We address the broader notion of a motion database and retrieval environment for practical motion editing sessions.

To concatenate the motion corpus efficiently together, Gleicher et al. [9] proposes a Snap-Together Motion system that processes a set of motion corpus to a set of short clips that can be concatenate to a continuous stream. The system builds a graph structure that facilities efficient planning of character motion. Kovar et al. [10] constructs a motion graph that consists of both original motion and automatically generated transitions to facilitate synthesizing walks on the ground. Both works involve retrieving similar character poses from the set of motion corpus. An efficient motion retrieval system will be beneficial to synthesizing more complex motions by maintaining a larger set of motion corpus.

Recently, some motion synthesis techniques are based on a manually annotated motion database [11,12]. A new motion is synthesized based on basic vocabulary of terms specified by the animator such as “walk”, “run” or “jump.” Thus, motion clips are hand-annotated before the synthesis. The new motion is reassembled by blending the annotated motion clips. Automatic annotation using Support Vector Machine (SVM) classifiers has also been investigated by Arikan et al. [12]. However, human intervention is necessary in the classification in order to guide the system to learn certain actions. Such a classifier can lead to the success of building a text-based motion retrieval database.

Other work involves the idea of “keyframe” or “keypose”. Pullen et al. [13] propose motion “texturing” based on the animation keyframes. Keyframes and curves are outlined by the animator and the motion capture data are mapped to the curve based on a matching process. Bevilacqua et al. [14,15] propose to extract keypose from motion capture data based on the magnitude of velocity. We provide better solution by exploring the information content present the transition between motion patterns. We then compare their results in the result section. Kim et al. [16] propose extracting motion rhythms based on the *zero-crossings* of the second derivative (acceleration) of each motion signal. A zero-crossing of the acceleration of motion signal could potentially mean a keypose. But their work addresses a very different problem. Their emphasis is on finding the motion beat. Potential keyposes outside the beat are not considered. This can pose problems in more generic types of motion sequences. For this it is important to rank the keyposes and attempt to find a measure of the significance in the change in motion pattern. Our work is based on using the mutual information metric in order to find all the potential keyposes in the motion with more emphasis on determining the significant keyposes.

The problem we are dealing with is analogous to the keyframe extraction in video indexing and retrieval system. Certain color histogram-based algorithms are developed for detecting video cuts [17-20] and fades [21-25]. Our method is similar to a recent video cut and fade detection algorithm by Z. Černeková et al. [26], which is based on entropy metrics of mutual information and joint entropy.

Entropy metric-based keypose extraction

Mutual information

The entropy of a discrete random variable X measures the information content or "uncertainty" of X. Let $A_x = \{a_1, a_2, ..., a_N\}$ be the outcome of X having probabilities $\{p_1, p_2, ..., p_N\}$ with $p_X = (x = a_i) = p_i$, $p_i \geq 0$ and $\sum_{x \in A_X} p_X(x) = 1$. The entropy is defined as:

$$H(X) = -\sum_{x \in A_X} p_X(x) \log p_X(x) \tag{1}$$

The joint entropy of two discrete random variables X and Y is given by:

$$H(X,Y) = -\sum_{x,y \in A_X, A_Y} p_{XY}(x,y) \log p_{XY}(x,y) \tag{2}$$

where $p_{XY}(x,y)$ is the joint probability density function. The mutual information measures the reduction in the uncertainty of X given the knowledge of Y:

$$I(X,Y) = -\sum_{x,y \in A_X, A_Y} p_{XY}(x,y) \log \frac{p_{XY}(x,y)}{p_X(x) p_Y(y)} \tag{3}$$

The mutual information can also be calculated by relating to the joint entropy of X and Y:

$$I(X, Y) = H(X) + H(Y) - H(X, Y) \tag{4}$$

The mutual information gives us a measure of association between X and Y. It measures the overlap information carried in X and Y. This is the overlap region of H(X) and H(Y). On the other hand, the joint entropy measures the unified information carried in X and Y, which is the union region of H(X) and H(Y).

Displacement Histogram and mutual information of displacement

The data we consider is the set of the global XYZ coordinates of the markers captured by a motion capture system. This format is currently generated by modern motion capture systems such as VICON. Let ($x_{i,f}$, $y_{i,f}$, $z_{i,f}$) be the global XYZ coordinates of marker i in frame f. We assume that the data does not contain missing samples. Missing marker data due to occlusion has to be fully reconstructed before the calculation and it can be performed as in [27].

We define vector $d_{i,f}$ be the displacement vector of marker i in frame f to the next frame, which is

$$d_{i,f} = \begin{pmatrix} dx_{i,f} \\ dy_{i,f} \\ dz_{i,f} \end{pmatrix} = \begin{pmatrix} x_{i,f+1} - x_{i,f} \\ y_{i,f+1} - y_{i,f} \\ z_{i,f+1} - z_{i,f} \end{pmatrix} \tag{5}$$

where $dx_{i,f}$, $dy_{i,f}$ and $dz_{i,f}$ correspond to the displacements in XYZ axis. $d_{i,f}$ forms an instantaneous velocity of marker i in frame f to the next frame.

The next step is to calculate the displacement histogram according to n number of discretization levels. Consider only the X coordinates. We choose the global maximum and minimum values of $dx_{i,f}$ as the range of the discretization. Let the displacement histogram $A_f^X = \{a_{f,1}^X, a_{f,2}^X, \ldots, a_{f,n}^X\}$ be the n discretized outcome of $dx_{i,f}$ in frame f having corresponding probabilities $B_f^X = \{p_{f,1}^X, p_{f,2}^X, \ldots, p_{f,n}^X\}$, where $p_{f,i}^X = a_{f,i}^X$ / total number of markers. Similarity, the joint probability of the discretized $dx_{i,f}$ between frame f and f+1 can be expressed as a $n \times n$ matrix $C_f^X(r,s)$, with $1 \le r \le n$ and $1 \le s \le$

n. Element (r, s) in $C_f^X(r,s)$ represents the probability of a marker having discretization level r in frame f and discretization level s in frame f + 1. Let I_f^X be the mutual information measure of the displacement from frame f to f + 1 for the X coordinates. We construct the mutual information measure I_f^X by applying equation (3) to the elements of vector B_f^X and matrix $C_f^X(r,s)$:

$$I_f^X = -\sum_{r=1}^{n}\sum_{s=1}^{n} C_f^X(r,s)\log\frac{C_f^X(r,s)}{B_f^X(r)B_f^X(s)} \tag{6}$$

We do the same formulation for the Y and Z coordinates in order to obtain the I_f^Y and I_f^Z. Hence, the total mutual information I_f of the displacement of frame f to f + 1 is

$$I_f = I_f^X + I_f^Y + I_f^Z \tag{7}$$

Keypose detection

A keypose involves a change of velocity (or a change of displacement vectors between frames) in the motion. A low mutual information I_f indicates a high probability of dissimilar velocity between frames, which indicates a high probability of occurrence of keypose. Figure 1 shows an example of mutual information plot of a motion sampled in 120Hz with n = 256 discretization levels. Lower I_f level indicates a relatively significant change of velocity and higher I_f level indicate a relatively similar movement. We propose to localize the mutual information in order to emphasize the local changes. We plot

$$\hat{I}_f = \frac{I_f}{E_w[I_f]} \tag{8}$$

where $E_w[I_f]$ = the mean of I_{f-w} to I_{f+w-1}. $E_w[I_f]$ forms a window always centered on frame f. We restrict the left boundary of the window to 1 when f-w < 1 and the right boundary to the maximum number of frames when f+w-1 exceeds the maximum number. Figure 2 shows the $\hat{I}_f$ plot of Figure 1 with w = 60 (total window size = 1 second). The vertical lines show the keyposes extracted as the low minima of $\hat{I}_f$.

There are several ways to pick the low minima of $\hat{I}_f$. The $\hat{I}_f$ curve can be trimmed horizontally with a threshold $\in$. Several local troughs will be isolated and the minimum value of each trough can be picked. Another way is to sort the $\hat{I}_f$ in an increasing order. We adaptively increase the threshold $\in$ until a particular number or density of troughs are reached. Only the lowest $\hat{I}_f$ point of each trough will be output as keypose. We may also consider to smooth the signal but this may potentially incur inaccurate detection of sharp mutual information changes. The example shown in Figure 2 is extracted with threshold $\in$= 0.9.

Results

Keypose extraction

Figure 3 shows the result of a dance motion. The motion is sampled at 120Hz. The localized mutual information $\hat{I}_f$ is calculated with 1 second windows size and 256 discretization levels. The keyposes are extracted with increasing threshold $\in$. The motion starts with a lean to the left and a lean to the right followed by a leap of the right leg and the left leg. The number of each snapshot in Figure 3 indicates the order of the extracted keyposes. In the beginning, the 2 most significant keyposes (largest change of motion) are extracted with low $\in$ (0.75). Two more keyposes (3 and 4) then appear when $\in$ increases to 0.90. The less significant keyposes (5, 6 and 7) are extracted with higher $\in = 0.95$. They are mainly the transit poses in between the significant poses. The significance of the keyposes decreases with increasing threshold level. The process stops upon a particular number or density of keyposes is reached, or a pre-defined maximum threshold is reached (1.0 is a reasonable number as we are interested in poses below average). Such snapshots of the keyposes give a brief summary of the motion events along the time.

Comparison with magnitude-velocity method

We try to compare our method with a magnitude-velocity method proposed by Bevilacqua [10]. The magnitude-velocity method calculates the magnitudes of the velocity of the markers at each frame to the next frame. All the magnitudes are summed up to form a 1-D signal. The local minima of the plot will be extracted as keyposes. In order to compare this method with our method, we try to localize the signal by the same 1-second window.

Our result shows that our mutual information method is more sensitive to directional changes than the magnitude-velocity method. Figure 4 shows the plot of our localized mutual information curve and the magnitude-velocity curve of another dance motion. Under the same threshold $\in = 0.90$, 22 keyposes are picked by our method (They are indicated by the vertical lines in the figure), and 15 keyposes are picked by the magnitude-velocity method. The figure shows that nearly all the keyposes picked by the magnitude-velocity are also picked by our method. We can name the extra keyposes detected by our method as transition keyposes

We further investigate the property of the transition keyposes in Figure 4. Figure 5 (upper) shows the zoomed view of the region X and Y. In region X, three keyposes are picked by the mutual information method. The keypose in interest is the circled one. The character rises his arms from the starting frame to the circled keypose and pulls back in the third keypose. The magnitude-velocity method fails to detect this change of direction since the motion is in continuous magnitude-deceleration. In region Y, the character is stretching his right leg. The uppermost point of where the character lifts his leg is extracted as keypose (circled in the figure). The leg is putting down after this pose. The magnitude-velocity method fails to detect this post again as the local maximum of the magnitude-velocity does not necessarily imply a change of direction.

In general, the transition keyposes detected by our method have no direct association to the shape of the magnitude-velocity curve nor its first derivative. Though some keyposes picked by our method meet the local maxima of the magnitude-velocity curve (the last keypose in region X and the circled keypose in region Y), not all the local maxima of the speed can simply imply changes of direction (as shown in the rest of Figure 4). The first keypose in Figure 5 also gives us as a good example that the transition keypose can be under a continuous speed deceleration.

Our method can also used to detect static part of the body. We can calculate the mutual information of the individual parts (2 limbs, 2 legs) of the body. The lower part of Figure 5 shows the result of the region X and Y. Two legs are relatively more static in region X and the left leg is relatively more static in region Y.

Motion database retrieval

Keypose extraction is a pre-processing step of motion database retrieval. It improves the matching efficiency by eliminating the similar poses near the keyposes. Liu et. al. [2] employs a clustering method to pick the keypose in the centroid for further elastic matching between two sequences. Our method can also be an alternative to the clustering method. Figure 6 shows the keyframe extraction with our method for an original motion and three other motions. The threshold $\in$ of the localized mutual information used is 1.00. 5 keyframes are extracted in each of the motion. Motion A and B are shown to be similar to the original motion. Motion C is shown to be dissimilar to the others. We can measure the similarity of keypose v and w in terms of their vector dot-products:

$$\theta(v,w) = \sum_{i=1}^{m} \frac{1}{m} \cos^{-1}\left(\frac{v_i \cdot w_i}{|v_i||w_i|} \right) \tag{9}$$

where v_i is the i-th vector defined between the markers of keypose v, and m is the total number of vectors available, which is the total number of markers minus 1. The vectors are expressed in terms of the local frame defined in the hip joint of the character in order to preserve the local orientation of the character. Equation (9) measures the average angle difference of the vectors between the two keyframes.

Figure 7 shows the average angle difference between the keyposes of the original motion and motion A, B, and C. Motion A is found to be very similar to the original motion since it has an overall angle error of $\pi/16$. Motion B is different at the beginning (about $\pi/8$) but very similar at the end of the motion sequence (about $\pi/16$). Motion C is found to be different overall as it has an error above $\pi/4$. The result shows that the method provides an efficient motion pattern comparison method independent of absolute time. For more precise motion comparison, the time gap between each keypose can be included in the distance function. We can add a penalty function for larger time gaps.

In summary, we have shown a practical method for our keypose extraction method for motion comparison. An open problem is to compare the motion with different number of keyposes in a motion retrieval system. The elastic matching proposed by Liu et. al. [2] can be a possible starting point for solving this problem since it considers the time-warping between keyposes.

Conclusion

We have presented a new keypose extraction method from motion captured data based on the mutual information metric. The method produces a list of keyposes ranked by their significance of directional change. The metric of the method is based on the mutual information of the displacements of the markers between frames. A lower mutual information level indicates a high probability of dissimilar displacements between frames. The main feature of our method is that it has a higher sensitivity to directional change than the magnitude-velocity method. This work provides the basis for another application such as music beat synchronization, motion browsing and a motion retrieval system. Our method is shown to be useful in extracting the “extreme” keyposes used in the motion similarity computations for a motion retrieval database.

References

[1] *VICON 8i*, VICON Motion Systems, Oxford Metrics Limited.

[2] F. Liu, Y. Zheung, F. Wu, and Y. Pan. 3D motion retrieval with motion index tree. In *Computer Vision and Image Understanding*, 92:265-284, 2003.

[3] Y. Li, T. Wang, and H. Y. Shum. Motion texture: A two-level statistical model for character motion synthesis. In *Proceedings of ACM SIGGGRAPH 2002*, Annual Conference Series, ACM SIGGRAPH, 465-472, 2002.

[4] A. Bruderlin and T. Calvert. Knowledge-driven, interactive animation of human running. In *Graphics Interface*, Canadian Human-Computer Communicates Society, 213-221, May 1996.

[5] L. Molina-Tanco and A. Hilton 2000. Realistic synthesis of novel human movements from a database of motion capture examples. In *Workshop on Human Motion 2000 (HUMO'00)*, 137-142, 2000.

[6] O. Arikan and D. A. Forsyth. Interactive motion generation from examples. In *Proceedings of ACM SIGGGRAPH 2002*, Annual Conference Series, ACM SIGGRAPH, 483-490, 2002.

[7] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of ACM SIGGGRAPH 2002*, Annual Conference Series, ACM SIGGRAPH, 491-500, 2002.

[8] M. Gleicher. Comparing Constraint-Based Motion Editing Methods. In *Graphical Models*, 63(2):107-134, 2001.

[9] M. Gleicher, H. J. Shin, L. Kovar, and A. Jepsen. Snap Together Motion: Assembling Run-Time Animation. In *Symposium on Interactive 3D Graphics 2003*, 181-188, April 2003.

[10] L. Kovar, M. Gleicher and F. Pighin. Motion Graphs. In *Proceedings of ACM SIGGGRAPH 2002*, Annual Conference Series, ACM SIGGRAPH, 473-482, 2002.

[11] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multi-dimensional motion interpolation. In *IEEE Computer Graphics and Applications*, 18(5):32-41, 1998.

[12] O. Arikan, D. A. Forsyth, and J. F. O'Brien. Motion Synthesis from Annotations. In *ACM Transactions on Graphics (TOG)*, 22(3):402-408, July 2003.

[13] K. Pullen and C. Bregler. Motion capture assisted animation: Texturing and synthesis. In *Proceedings of ACM SIGGGRAPH 2002*, Annual Conference Series, ACM SIGGRAPH, 501-508, 2002.

[14] F. Bevilacqua, J. Ridenour, and D. J. Cuccia. 3D motion capture data: motion analysis and mapping to music. In *Proceedings of the Workshop/Symposium on Sensing and Input for Media-centric Systems 2002*, Santa Barbara CA, 2002.

[15] F. Bevilacqua, L. Naugle and I. Valverde. Virtual dance and music environment using motion capture. In *Proceeding of IEEE Multimedia Technology And Applications Conference*, Irvine CA, 2001.

[16] T. Kim, S. I. Park, and S. Y. Shin. Rhythmic-motion synthesis based on motion-beat analysis. In *ACM Transactions on Graphics (TOG)*, 22(3):392-401, July 2003.

[17] G. Ahanger and T. D. C. Little. A survey of technologies for parsing and indexing digital video. In *Journal of Visual Communication and Image Representation*, 7(1):28-43, 1996.

[18] A. Dailianas, R. B. Allen, and P. England. Comparison of automatic video segmentation algorithms. In *Proceedings, SPIE Photonics East'95: Integration Issues in Large Commercial Media Delivery Systems*, Philadelphia, 2615:2-16, Oct. 1995.

[19] N. V. Patel and I. K. Sethi. Video shot detection and characterization for video databases. In *Pattern Recognition*, 30(4):583-592, April 1997.

[20] S. Tsekeridou and I. Pitas. Content-based video parsing and indexing based on audio-visual interaction. In *IEEE Transactions on Circuits and Systems for Video Technology*, 11(4):522-535, 2001.

[21] M. S. Drew, Z.-N. Li, and X. Zhong. Video dissolve and wipe detection via spatio-temporal images of chromatic histogram differences. In *Proceeding of IEEE International Conference on Image Processing (ICIP 2000)*, 3:909-932, 2000.

[22] R. Lienhart. Reliable dissolve detection. In *Proceeding of SPIE Storage and Retrieval for Media Databases 2001*, 4315:219-230, Jan. 2001.

[23] R. Lienhart and A. Zaccarin. A system for reliable dissolve detection in video. In *Proceeding of IEEE International Conference on Image Processing (ICIP 2001)*, Thessaloniki, Greece, Oct. 2001.

[24] Y. Wang, Z. Liu, and J.-Ch. Huang. Multimedia content analysis using both audio and visual clues. In *IEEE Signal Processing Magazine*, 17(6):12-36, Nov. 2000.

[25] R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying production effects. In *ACM Journal of Multimedia Systems*, 7:119-128, 1999.

[26] Z. Černeková, C. Nikou, and I. Pitas, Entropy metrics used for video summarization. In *Proceedings of the 18th Spring Conference on Computer Graphics*, Buderice, Slovakia, 73-82, 2002.

[27] K. Kurihara, S. Hoshino, K. Yamane and Y. Nakamura. Optical motion capture system with pan-tilt camera tracking and realtime data processing. In *Processings of 2002 IEEE International Conference on Robotics and Automation*, 1241-1248, May 2002.

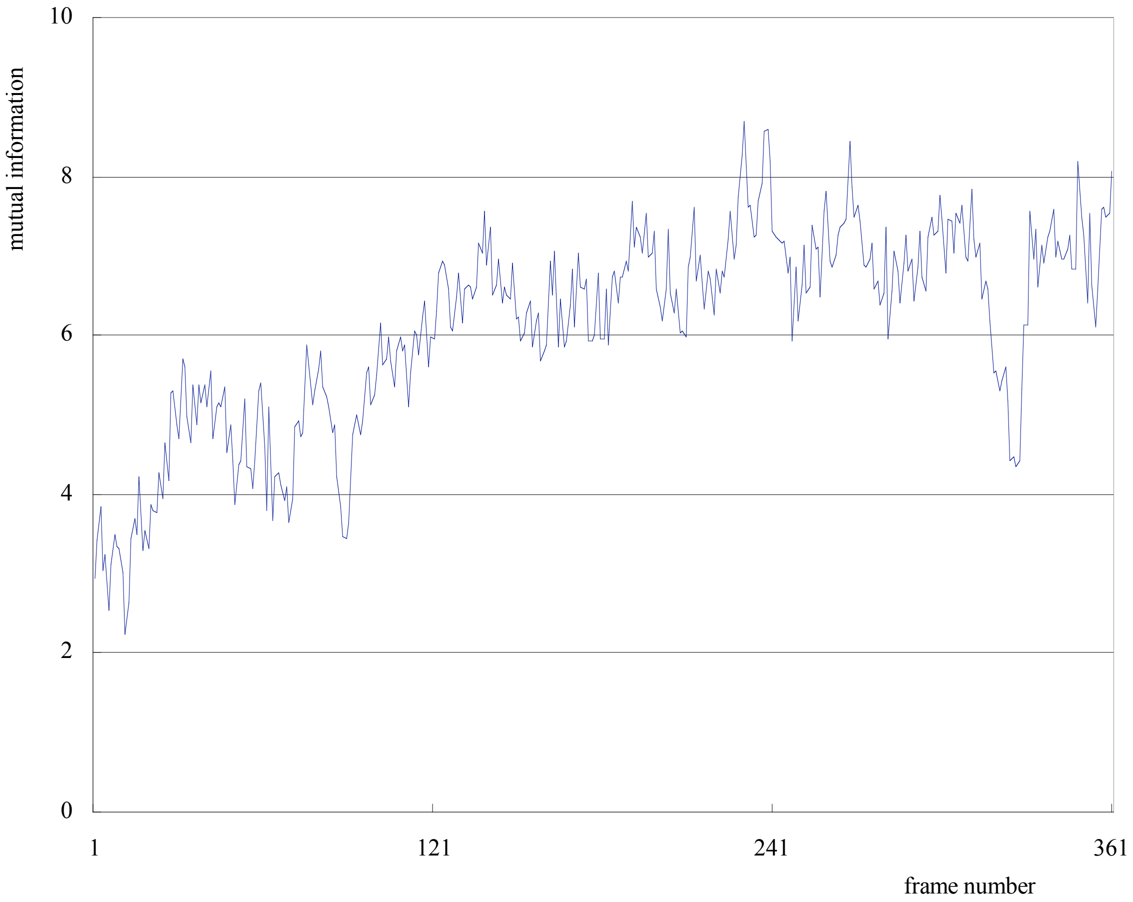

Figure 1: The mutual information I_f of an example motion sampled at 120 Hz. The displacement discretization level n = 256.

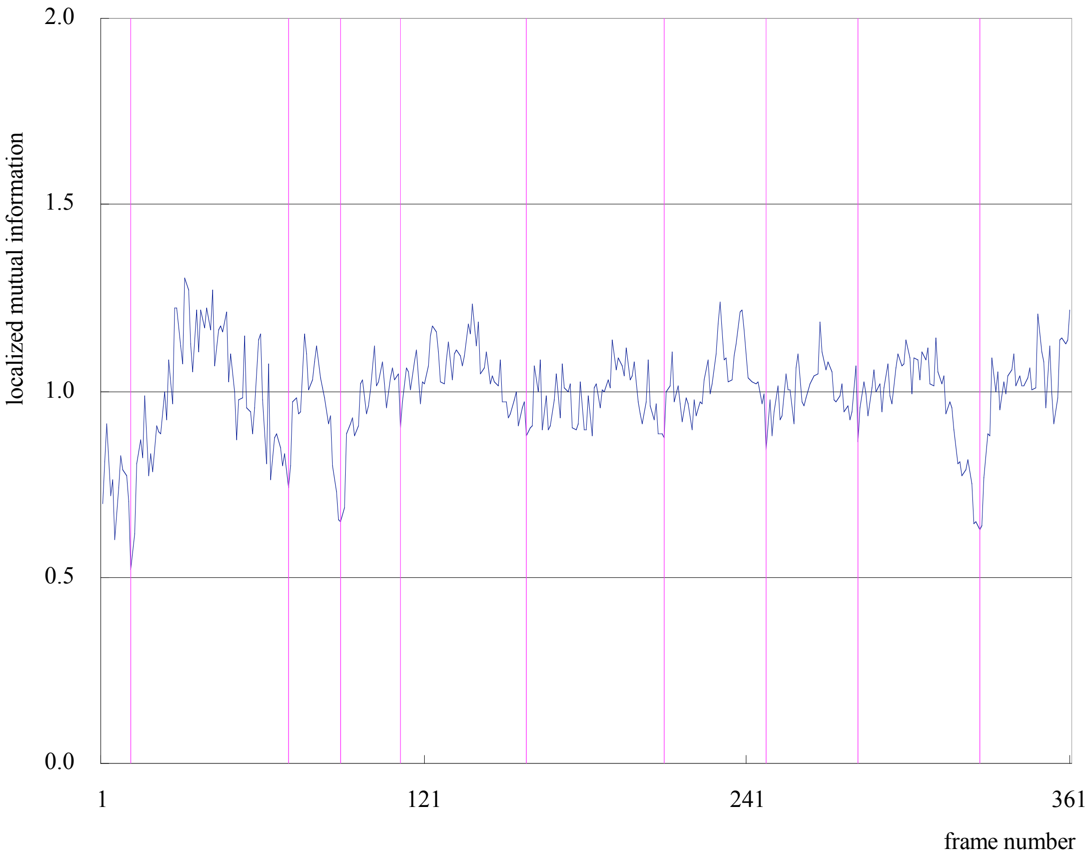

Figure 2: The localized mutual information If of the motion in Figure 1. The window size is 1 second. The vertical lines indicate the position of the picked keyposes with threshold $\in = 0.9$.

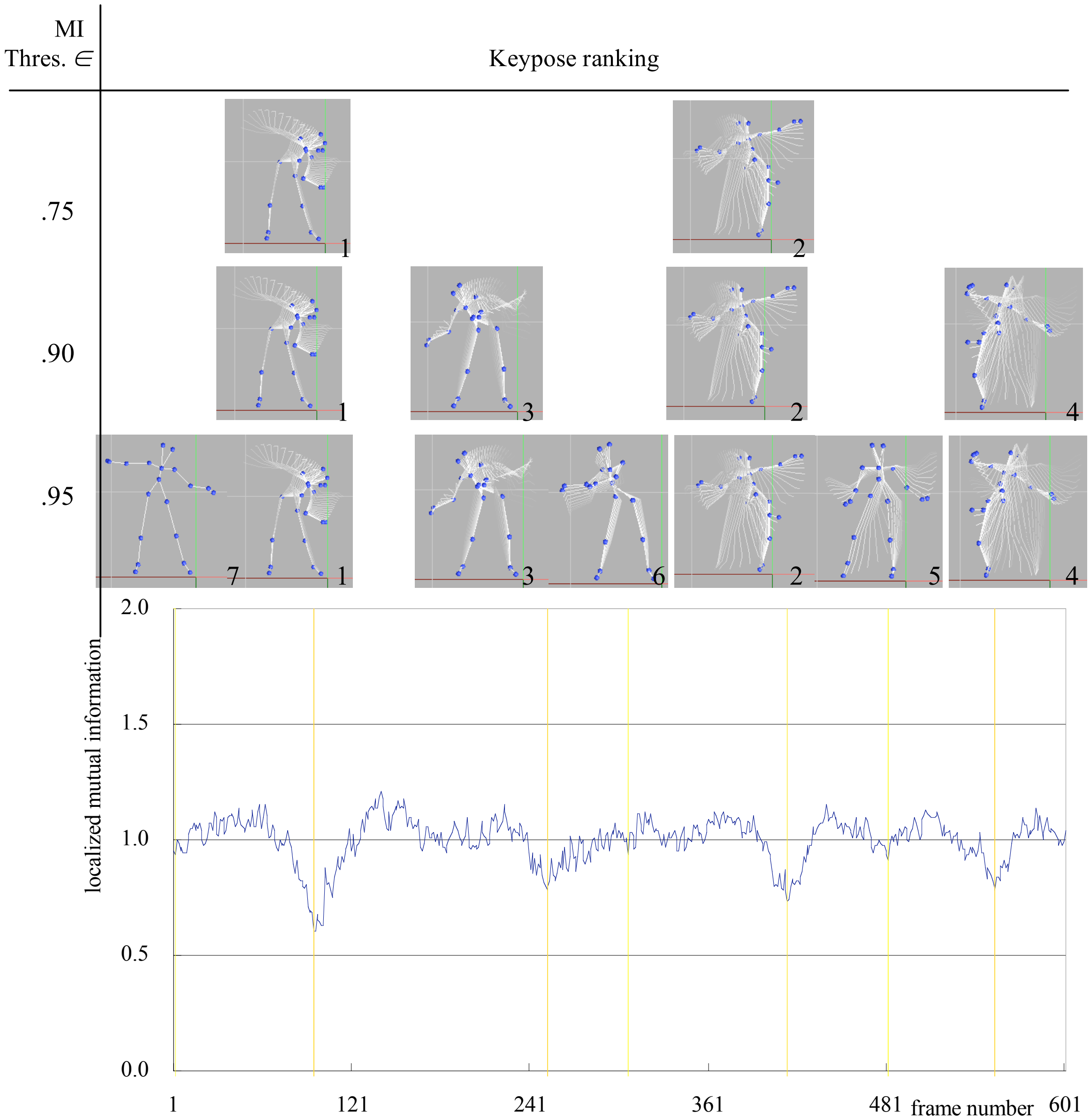

Figure 3: The picked keyposes of a dance motion with increasing threshold ∈. The numbers on the snapshots show the order (significance) of the picked keyposes. The vertical lines indicate the position of the keyposes with threshold ∈= 0.95.

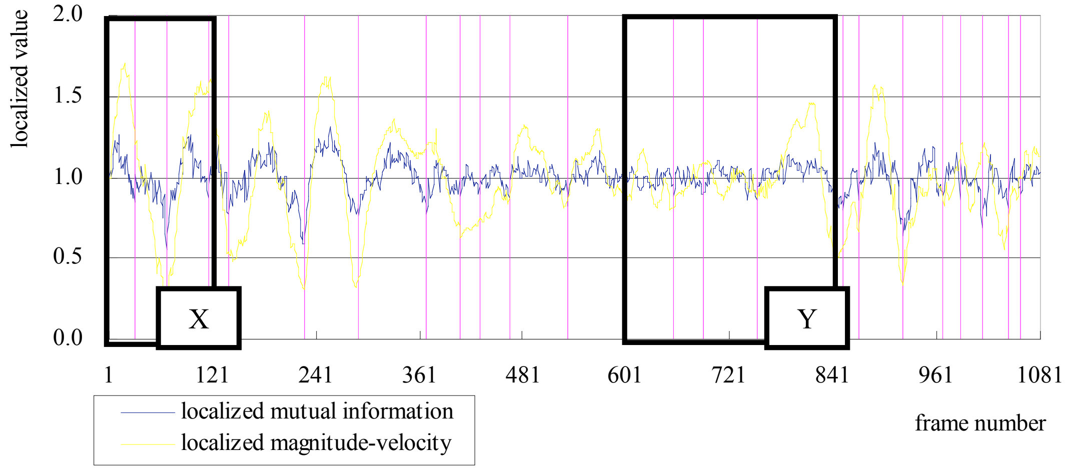

Figure 4: The localized mutual information and the localized velocity of another dance example. The vertical lines indicate the position of the picked keyposes with threshold $\in = 0.90$.

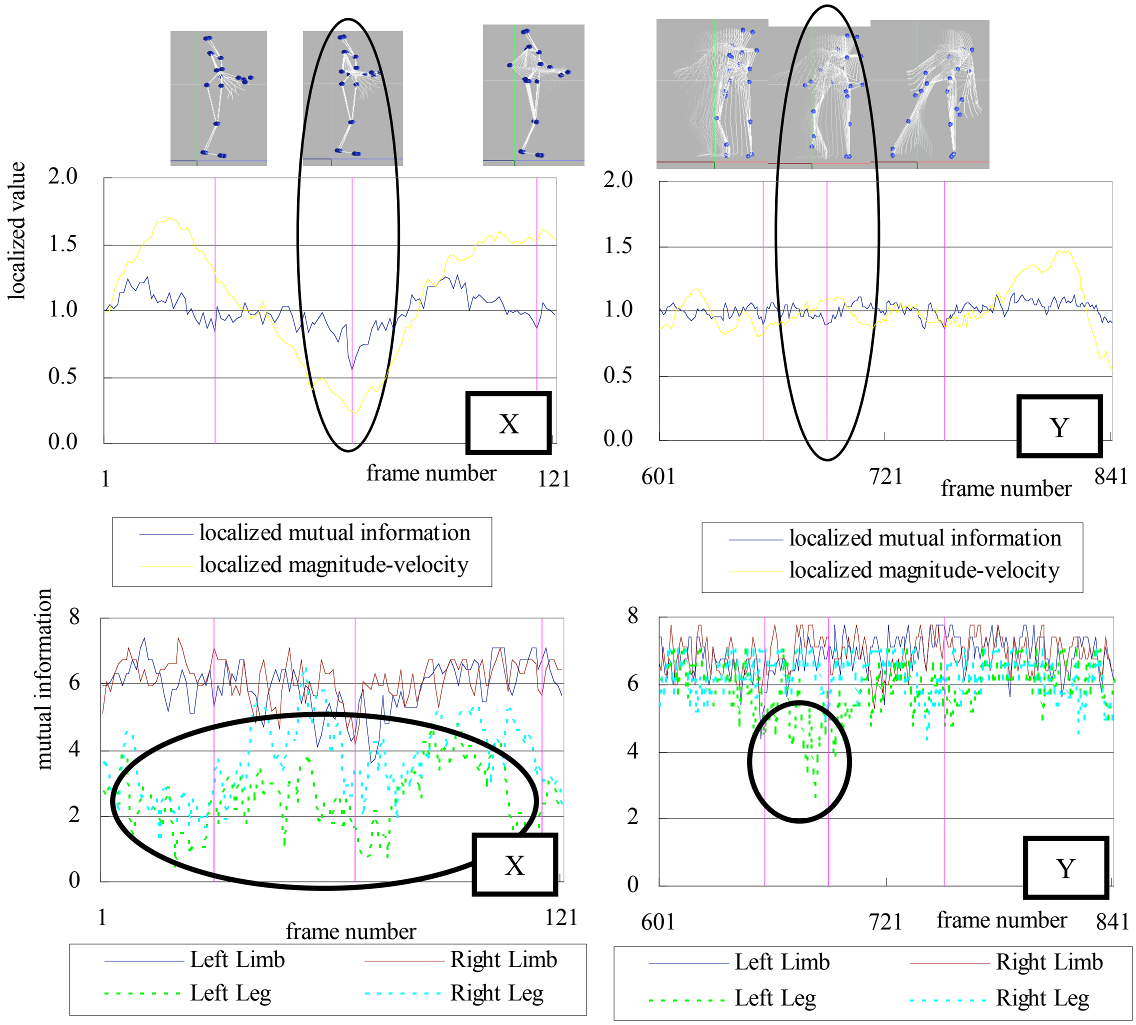

Figure 5: The zoomed figures (upper) of the region X and Y of the figure 4. The mutual information method picks keyposes with change of direction. The corresponding mutual information plots of the legs and limbs are given below. Relatively static motion of the legs is detected.

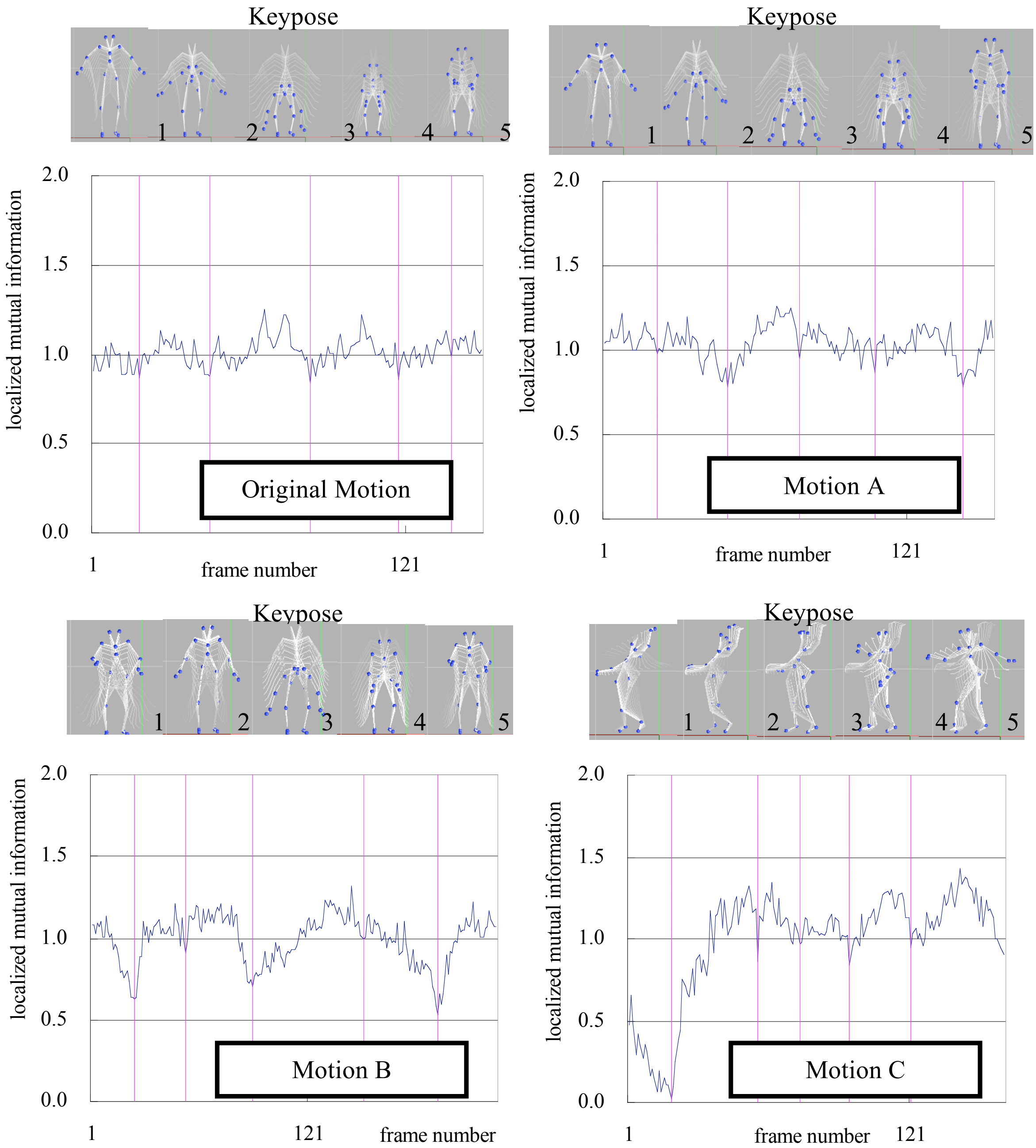

Figure 6: The mutual information and the 5 extracted keyposes of 4 motions. The keyposes are extracted with MI threshold $\in = 1.00$. Motion A and B are similar to the original motion. Motion C is dissimilar to the others.

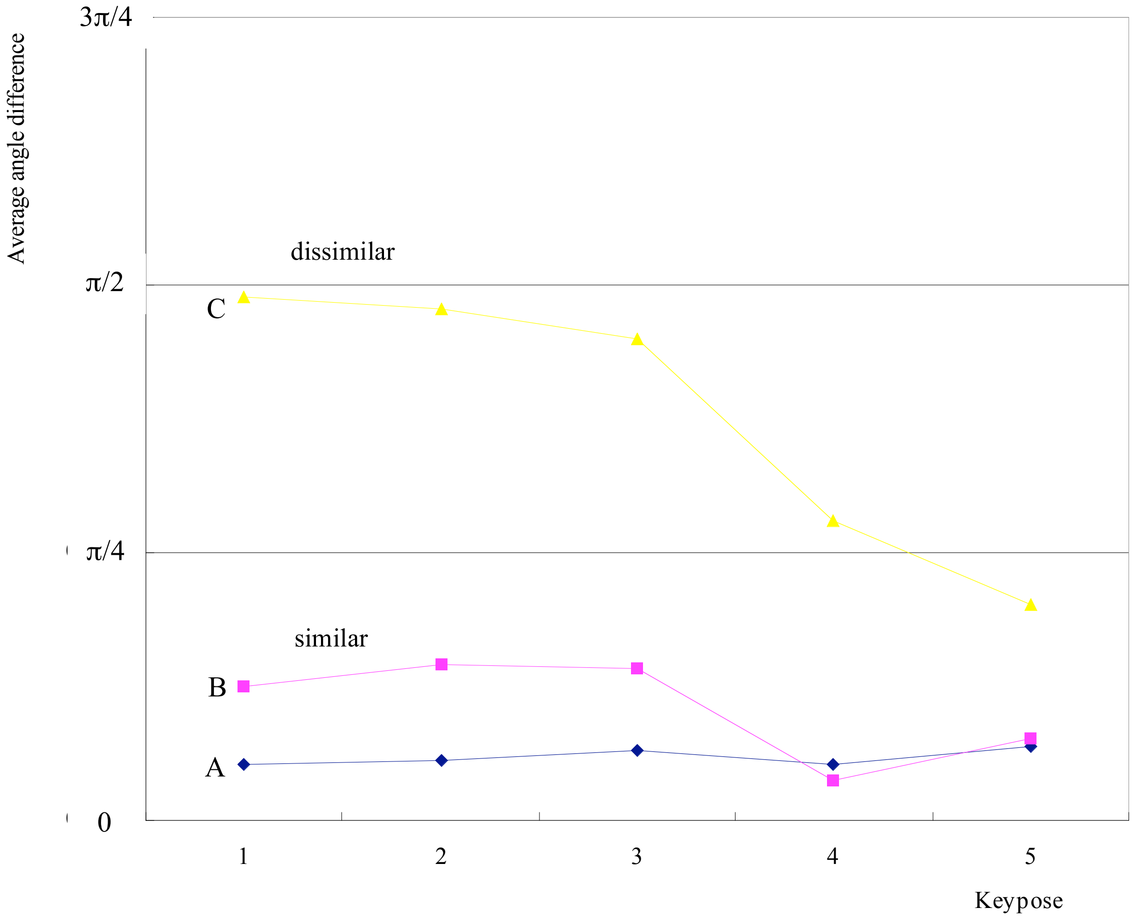

Figure 7: The average angle difference between the original motion and the motion A, B and C of the 5 keyposes.