8

A MOTION RECOGNITION METHOD BY USING PRIMITIVE MOTIONS

Ryuta Osaki†, Mitsuomi Shimada † and Kuniaki Uehara ‡
† *Department of Computer and System Engineering,*
Graduate School of Science and Technology, Kobe University
{ryuta, shimada}@ai.cs.kobe-u.ac.jp
‡ *Research Center for Urban Safety and Security, Kobe University*
uehara@kobe-u.ac.jp

Abstract Contents-based retrieval of multimedia information has been investigated in several research projects. In this paper, we will focus on an automatic indexing method for human motion data. We convert a motion data, which is represented as time series of 3-D position, into a symbol sequence. We call this method as conversion automatic indexing. The automatic indexing is performed in a pattern matching approach. Reference patterns are necessary for pattern matching, so that we will propose two methods to define primitive motions in order to make reference patterns. The first method divides motion data into segmental motion data by detecting the change of motion speed. The second method classifies segmental motions such that similar segmental motions are gathered in the same cluster. In order to evaluate the similarity between two segmental motions, we use the Dynamic Time Warping (DTW) method because each segmental motion takes different time length even if the same person performed the same motions. Motion data can be converted into a symbol sequence which represents a sequence of primitive motions. Then, Continuous Dynamic Programming (CDP) method is used to recognize contents of motion. CDP is one of the extensions of DTW. It makes us possible to recognize a motion with ease even if it is complex.

Keywords: primitive motion, motion database, motion recognition, contents-based retrieval, dynamic time warping

1. INTRODUCTION

The necessity to deal with human motion data on computer is growing in the field of movies, video-games, animations and so on [Stuart et al., 1998]. Storing motion data is also important so as to reuse them. Since

H. Arisawa et al. (eds.), *Advances in Visual Information Management*

motion data is difficult to be described by using key words, it is better to use motion data itself as query to access the database. This is what is called contents-based retrieval. A database system for contents-based retrieval accepts vague queries and it performs a best-match search to find data that are likely to be most relevant to the queries. Contents-based retrieval is based on features that contain clues about the content of data. These features are generated by an automatic indexing process.

In this paper, we will focus on automatic indexing method of motion data [Osaki et al., 1999]. The method is based on speech recognition technique, because motion is similar to speech in the sense that they are time series data. Two major approaches in the research of speech recognition have been proposed, probabilistic model based approach [Kuhn et al., 1995] and pattern matching approach [Sakoe et al., 1971]. In both approaches, speech is phonetically transcribed to phonemes, and phonemes are compared to reference patterns or models. The problem is that primitive motions have not been studied to employ the speech recognition methods. Therefore, we have to develop both a method to extract primitive motions and automatic indexing method.

After primitive motions are extracted, each of them is represented by a symbol. That is, motion data can be converted from a huge amount of position data into a small number of symbols. This makes it easy to analyze a motion data but still leaves two problems: One is a wide variety of motion patterns and the other is the indexing error due to the noise that occurs when motion data is converted into a symbol sequence. To solve these problems, we use Continuous Dynamic Programming (CDP) [Hayami et al., 1984]. CDP is a method for connected word recognition. It uses a dynamic programming approach to align the time series and a specific reference pattern so that some distance measure is minimized. Since the time axis is stretched (or compressed) to achieve a reasonable fit, a reference pattern may match a wide variety of actual time series. By using CDP, connected motion can be recognized and the indexing error is reduced by compressed time axis.

2. RELATED WORKS

Many researchers focus on the analysis of human motion due to its variety of applications. For example, Rohr [Rohr, 1994] proposed a model-based approach for the recognition of walkers. He used a 3D-model to represent the human body and applied a Kalman filter to estimate the model parameters. Gavrila et al. [Gavrila et al., 1995] proposed a model-based tracking and recognition system. Their system contains two components: (1) taking real image sequences acquired from

Figure 1 An example of motion capture system.

multiple views and recovering the 3D body pose at each time instant, (2) representing and recognizing of human movement patterns.

Generally speaking, motion recognition method can be divided into two phases: retrieving an appropriate motion data and recognizing the content of motion data. Most of researchers stress the former phase and they try to retrieve 3-D motion information from 2-D images. But, it is difficult because images are influenced by many conditions such as background image, person, clothes and so on. In addition, if the movement is more complex, the extraction of motion information becomes more difficult because of serious occlusion. Thus, motions which can be recognized by humans are limited to simple motions, like waving hands, rising one's hand, walking and so on.

On the contrary, we are interested in the latter phase. We use optical motion capture system to record time series of 3-D motion data. Figure 1 shows an example of performer wearing 24 markers which reflect infrared ray on each joint. Each motion data is represented as 3-D locations of each joint.

3. EXTRACTION OF PRIMITIVE MOTIONS

First of all, our system divides all the motion data into segmental motions [Pavlidis et al., 1974][Das et al., 1998] by detecting the change of motion speed. Next, it classifies similar segmental motion data into the same cluster by using the nearest neighbor algorithm with DTW algorithm for distance function. We call each cluster primitive motion.

We have to find breakpoints where primitive motion data are combined. Generally, every motion can be analyzed as follows: start motion,

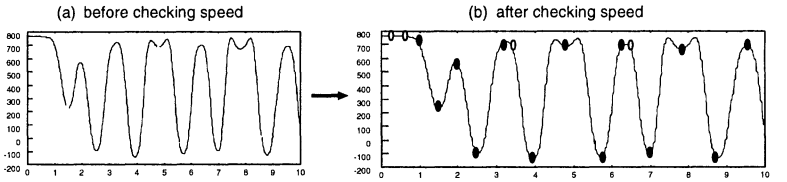

Figure 2 An example of checking speed.

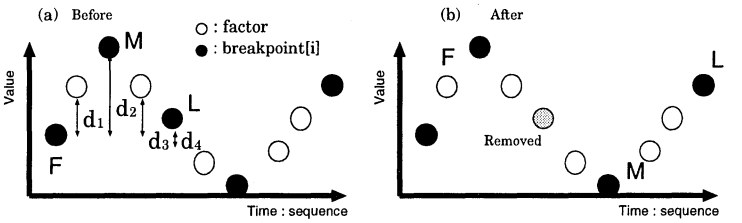

Figure 3 Discernment of breakpoints.

transitional motion, and stop motion. Start motion represents a change from zero speed to a movement, transitional motion represents a change in the speed and stop motion represents a change from a movement to zero speed. Thus, each change in the speed of a body part can be considered as a breakpoint. However, vibration, such as hand shaking, may be detected as breakpoints by mistake. For this reason, the algorithm calculates the variance between all the candidate points. If the variance is small, it means that vibration occurred between the candidate points. These points should be neglected, and removed from the candidate points. This process is executed in x, y, and z-axis respectively.

Figure 2 shows an example of this process. Figure 2 (a) is an original data, and (b) is a result of checking speed. In Figure 2 (b), a circle represents a breakpoint. Almost the breakpoints are proper, but some of them are not. For example, four breakpoints (white points in Figure 2(b)) are detected by mistake. They should not be considered as breakpoints. To remove such breakpoints, the other process is executed to evaluate the spatial relationship view. Figure 3 shows an example which represents a motion data in one dimension for simplicity.

The motion data in Figure 3 (a) includes ten points, in which five of them are candidates to be breakpoints (black points). Figure 3 (b) represents the breakpoints which were detected by our algorithm. One of the original candidate ("L" in Figure 3 (a)) was removed in Figure 3 (b).

The algorithm works as follows: let us consider the sequence of three candidate breakpoints (i.e. "F," "M" and "L"). Distances from "F" to all candidate points are calculated in order to find the most distant point from "F." In the same way, the most distant point from "L" is found. For example, "d_i" ($1 \leq i \leq 4$) represent the distances from "F" to four candidate points, and "d_2" is the most distant point from "F." If neither of most distant points from "F" and "L" are "M," this means slight vibration occurred in the motion data. In other words, "M" is unsuitable for the boundary of primitive motions. Thus, "M" is removed from the candidates. And the next three candidate points are considered until all the candidate points are checked. Finally, we can obtain Figure 3 (b) as the result.

Dynamic Time Warping. DTW is a method that was developed in the field of speech recognition. DTW calculates a similarity of the discrete time series. We can recognize whether the discrete time series are the same or not in accordance with the similarity. There is an example. Assume two discrete time series A and B are given as follows:

$$\begin{aligned} A &= a_1, \ldots, a_i, \ldots, a_M \\ B &= b_1, \ldots, b_j, \ldots, b_N \end{aligned} \tag{1}$$

where a_i and b_j are the i th and j th points of given A and B. The similarity between A and B is given the following formulas (2) and (3).

$$d(i,j) = \sqrt{(a_i - b_j)^2} \tag{2}$$

$$\begin{aligned} S(i,j) &= d(i,j) + min\{S(i-1,j), S(i-1,j-1), S(i,j-1)\} \\ DTW(A,B) &= S(M,N),\ S(1,1) = d(1,1) \end{aligned} \tag{3}$$

Although this similarity function is enough for 1 dimensional data, such as speech recognition, we must extend the function for 3 dimensional data for motion recognition. Then we introduce formula (4) instead of formula (2),

$$d(i,j) = \sqrt{(a_{x_i} - b_{x_j})^2 + (a_{y_i} - b_{y_j})^2 + (a_{z_i} - b_{z_j})^2} \tag{4}$$

where a, b, i and j are the same in formula (1). x, y and z represent the axis for 3-D space. In order to deal with 3-D position data as spatial time series, it is necessary to evaluate x, y and z position data simultaneously.

4. AUTOMATIC INDEXING

There are two problems that make it difficult to analyze motion data:

1 Boundary detection of the motion data.

2 Running time of the analysis algorithm on huge data, such as motion data.

Therefore, we convert a motion data into symbol sequences in terms of primitive motions. We call this conversion "automatic indexing."

Automatic indexing is executed with the following procedure. Suppose that we have F varieties of motion data $M_1, \ldots, M_f, \ldots, M_F$ as references. Each motion M_f represents a primitive motion, such as throwing a ball, jumping and so on. M_f consists of the 3-D time series data for each body part, m_p^f, where p is the identifier of the body part.

First of all, we divide each m_p^f into $s_p^f 1, \ldots, s_p^f n$ segments and classify them into $C_p^1, \ldots, C_p^K$ sets, where k is the number of clusters for each body part. For each cluster C_p^k we assign a symbol a_p^h, and each m_p^f will be over the alphabet $\sum = \{a_p^1, \ldots, a_p^h\}$. The indexed m_p^f is obtained by looking for the cluster $C_p^{j(i)}$ such that $C_p^{j(i)}$ contains the most similar data to $s_p^f i$, and by replacing $s_p^f i$ with corresponding symbol $a_p^{j(i)}$. Each $a_p^{j(i)}$ is the index of $C_p^{j(i)}$. Then,

$$m_p^f = a_p^{j(1)} a_p^{j(2)} \ldots a_p^{j(n)} \tag{5}$$

We call $a_p^{j(1)} a_p^{j(2)} \ldots a_p^{j(n)}$ a symbol sequence, then a pair of m_p^f and the symbol sequence is stored in the database as reference patterns.

We also divide the input motion data M_{input} into $s_p^{input} 1, \ldots, s_p^{input} n$ segments. The symbol sequences of m_p^{input} are also obtained by:

1 finding $s_p^f k_t$ which is most similar to $s_p^{input} t$, such that $s_p^f k_t \in C_p^{j(k_t)}$. The similarity is given by DTW function.

2 using the corresponding symbol $a_p^{j(k_t)}$ to $C_p^{j(k_t)}$.

Thus m_p^{input} is converted into following sequence,

$$m_p^{input} = a_p^{j(k_1)} a_p^{j(k_2)} \ldots a_p^{j(k_n)} \tag{6}$$

Then symbols are assigned to any $s_p^{input} t$, and input motion data is also converted into the symbol sequence.

5. MOTION RECOGNITION

Motion recognition is carried out by comparing those symbol sequences of input with those of the reference patterns. If the sequences

contains complex motion, it is necessary to find boundaries and to match segmental patterns with reference patterns. However, there are many candidates of a boundary and a decision of boundary influences to matching process. To solve this problem, we will introduce continuous DP (CDP) to analyze symbol sequences.

One of the features of CDP is that three operations of motion boundary detection, nonlinear time alignment, and recognition are performed simultaneously; thus, recognition errors due to errors in motion boundary detection or due to time alignment errors are not possible. The algorithm is forced to match complete motions, and as a result of this, the motion boundaries are determined automatically.

Assume that two discrete time series A and B are given as formula (1). First of all, CDP calculates similarity S_i between any sub-series $a_1, \ldots, a_i$ in A and B by the following formula (7):

$$\begin{aligned} S_i(1,1) &= d(1,1) \qquad (7)\\ S_i(l,m) &= min \begin{cases} S_i(l-2,m-1) + 2S_i(l-1,m) + d(l,m) & \text{(a)} \\ S_i(l-1,m-1) + 2d(l,m) & \text{(b)} \\ S_i(l-1,m-2) + 2S_i(l,m-1) + d(l,m) & \text{(c)} \end{cases} \end{aligned}$$

where $d(l,m)$ is a distance between a_l and b_m. $S_i(i,m)$, the sum of $d(l,m)$, is normalized by $L_i(i,m)$ to remove the fluctuation caused by difference of time series length in any $a_1, \ldots, a_i$. Finally the similarity is given as:

$$CDP(A,B) = min\left\{\frac{S_i(i,N)}{L_i(i,N)}\right\} \qquad (8)$$

$L_i(i,m)$ is obtained simultaneously with S_i by using the following formula (9):

$$\begin{aligned} L_i(1,1) &= 0 \\ L_i(l,m) &= \begin{cases} L_i(l-2,m-1) + 3 & (if\ \text{(a)}) \\ L_i(l-1,m-1) + 2 & (if\ \text{(b)}) \\ L_i(l-1,m-2) + 3 & (if\ \text{(c)}) \end{cases} \qquad (9) \\ EndPoint &= argmin\left\{\frac{S_i(i,N)}{L_i(i,N)}\right\} \qquad (10) \end{aligned}$$

For the symbol sequence, we give the distance between a_l and b_m as follows:

$$d(l,m) = \begin{cases} 0 & (a_l = b_m) \\ 1 & (a_l \neq b_m) \end{cases} \qquad (11)$$

Figure 4 shows an example of motion recognition with CDP. Assume that a symbol sequence X of a body part is given as input, and we have

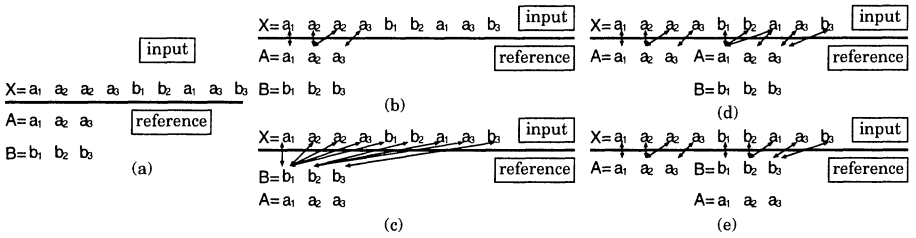

Figure 4 An example of CDP.

Table 1 Clustering results.

Segment name	Number of segments	Number of clusters	Number of errors	percentage
Left wrist	466	203	27	86.7
Right wrist	469	187	23	87.7
Left elbow	395	100	19	81.0
Right elbow	397	107	18	82.4
Left knee	206	30	12	60.0
Right knee	195	25	9	64.0
Left foot	258	50	20	60.0
Right foot	248	50	17	64.0

two reference symbol sequences A and B for the same body part (Figure 4 (a)). CDP calculates the similarities $CDP(X, A)$ and $CDP(X, B)$ (Figure 4 (b) and (c)). Then, it chooses $min\{CDP(X, A), CDP(X, B)\} = CDP(X, A)$ as A is the first pattern in X, and detects the motion boundary between a_3 and b_1 in X. In the next step, it chooses $min\{CDP(X, A), CDP(X, B)\} = CDP(X, B)$ as B is the second pattern in X (Figure 4 (d) and (e)). Finally, it gives the recognition result, that is, $X = AB$.

6. EXPERIMENTAL RESULTS

Our motion data are based on 24 different aerobic exercise. In this section, we show three experiments, clustering results, automatic indexing results and motion recognition results.

6.1. CLUSTERING RESULTS

Table 1 shows that there are some cases that classification errors occurred, and a large number of clusters are constructed. Assume two human performers with different length of arms. Even if they try to

Table 2 Motion recognition results.

Segment name	Correctly recognized segments	Total number of segments	Accuracy
Left wrist	92	106	86.8
Right wrist	83	104	79.8

perform the same motion, their left wrists take different courses. This is because our system incorrectly speculates that these two motion data are different since the similarity of the motion is evaluated by the formula (4).

6.2. MOTION RECOGNITION RESULTS

Table 2 shows the result of connected motion recognition of two body parts. In Table 2, accuracy is the percentage of correctly recognized segments from total number of segments of the input motion data. CDP found the boundaries for the most of reference symbol sequences on the input symbol sequences. The recognition results are as high as almost 80% accuracy although "noise" is included in the input motion data. That is to say, CDP can reduce the influence of "noise" and works the recognition process effectively, but more improvements are needed for higher accuracy.

Recognition errors are classified into two groups. Those errors are caused by CDP and the reference symbol sequences, the reference models. The reason is as follows: CDP detects the boundaries step by step. CDP starts next recognition from just after the end point, which is decided by the previous matching as shown in Figure 4 (d). If the boundary is not correctly detected, the next matching will be resulted in fail. In other words, if previous matching is detected wrong, it will affects the next matching result.

The reference models are also important. In this experiment, the references are unique symbol sequences for each motion, but widely fitable reference models are required for such a broad variety of data, motion data.

7. CONCLUSION

In this paper, we suggested the method to extract primitive motions from human motion data, including results by using this method. This paper proposed that human motions can be decomposed and the system can extract them as primitive motions. We also suggested a motion

recognition algorithm in terms of symbol sequences. Those sequences represents the motions as the sequences of extracted primitive motions. Our method can easily convert large time series of 3-D motion data into a small number of strings called symbol sequences. However, results shown in the Section 6 have to be improved to be applied for practical contents-based retrieval of motion data.

Acknowledgments

This work was partially supported by the grant-in-aid for scientific research on priority area "Discovery Science" from the Japanese Ministry of Education, Science, Sports and Culture.

References

[Osaki et al., 1999] Osaki, R., Shimada, M. and Uehara, K., Extraction of Primitive Motion for Human Motion Recognition. *In Proc. of the 2nd International Conference on Discovery Science.* Lecture Notes in Artificial Intelligence, Vol. 1721, pp. 351–352, Springer (1999).

[Stuart et al., 1998] Stuart, J. and Bradley, E., Learning the Grammar of Dance. *Proc. of 15th ICML*, pp. 547–564 (1998).

[Kuhn et al., 1995] Kuhn, T., Niemann, H. and Schukat-Talamazzini, E. G., Ergodic Hidden Markov Models and Polygrams for Language Modeling, *Proc. of ICASSP'94*, pp.357–360 (1994).

[Sakoe et al., 1971] Sakoe, H. and Chiba, S., Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Trans. Acoust., Speech, Signal Processing.* Vol. ASSP-26, pp. 43–49 (1978).

[Hayami et al., 1984] Hayami, S. and Oka, R., Experimental Studies on the Connected Words Recognition Using Continuous Dynamic Programming. *IEICE Trans.* Vol. J-67-D, No. 6, pp. 677–684 (1984).

[Rohr, 1994] Rohr, K., Toward Model-Based Recognition of Human Movements in Image Sequence. *CVGIP : Image Understanding*, Vol. 59, No. 1, pp. 94–115 (1994).

[Gavrila et al., 1995] Gavrila, D. M. and Davis, L. S., Towards 3-D Model-Based Tracking and Recongition of Human Movement: a Multi-View Approach. *Proc. of International Workshop on Automatic Face and Gesture Recognition.* (1995).

[Pavlidis et al., 1974] Pavlidis, T. and Horowitz, S. L., Segmentation of Plane Curves. *IEEE Transactions on Computers*, Vol. C-23, No. 8, pp. 860–870 (1974).

[Das et al., 1998] Das, G., Lin, K., Mannila, H., Renganathan, G. and Smyth, P., Rule Discovery from Time Series. *Proc. of 4th In-*

ternational Conference on Knowledge Discovery and Data Mining, pp. 16–22 (1998).

Biographies

Ryuta Osaki He is a graduate student of Graduate School of Science and Technology, Kobe University. He is a student member of the Information Processing Society of Japan.

Mitsuomi Shimada He is a graduate student of Graduate School of Science and Technology, Kobe University.

Kuniaki Uehara Since 1997, he has been a professor of the Research Center for Urban Safety and Security and Department of Computer and Systems Engineering of Kobe University. His current research interests include machine learning, natural language processing, and intelligent software engineering. He is a member of the Information Processing Society of Japan, Japan Society for Software Science and Technology, and Japan Society of Artificial Intelligence and AAAI.