DATA-DRIVEN AUTOMATIC GENERATION OF DECISION TREE FOR MOTION RETRIEVAL WITH TEMPORAL-SPATIAL FEATURES

JIAN XIANG, YUE-TING ZHUANG, FEI WU

College of Computer Science, Zhejiang University, Hangzhou 310027, China
E-MAIL: xiang_bj@zju.edu.cn, yzhuang@cs.zju.edu.cn, wufei@zju.edu.cn

Abstract:

Along with the development of Motion Capture technique, more and more 3D motion libraries become available. In this paper, a novel approach is presented for motion retrieval based on data-driven decision tree with 3D temporal-spatial features. First 3D temporal-spatial features of each human joint are extracted with the help of keyspace. Since the gotten features of each joint are independent, data-driven decision tree is automatically constructed to reflect the influence of each point during the comparison of motion similarity. Experiment results show that the approaches are effective for motion data retrieval.

Keywords:

Motion Capture; 3D Temporal-Spatial; Data Driven; Decision Tree; Motion Retrieval

1. Introduction

In recent years, motion capture technique and realistic human motion data have been widely used in computer games, animation production and medical simulations [1], [2], etc. For making the best of motion data in the large-scale 3D motion database, an efficient motion data retrieval technique is needed to support motion data processing such as motion morph, motion edit and synthesis, etc.

In order to retrieve motion data precisely and efficiently, following challenges should be focused on:

1) Since each motion is a harmonic combination of sub-motion of all the body joints, an efficient description mechanism of the motion feature is required during retrieval. The description mechanism has great effect on the motion data processing.

2) Because the dimension of motion feature extracted is very high, the distances between each two motion data are almost the same and cannot be discriminated according to central limit theorem, which called "Curse of dimensionality"[3].

3) In essence, motion is a kind of time series data, measure of the similarity for time series data is very difficult.

To meet above-mentioned challenges, firstly, temporal-spatial features are defined in this paper, which describes 3D space relationship of each joint. Comparing with the aforementioned motion features [4, 5, 6, 7] made up of 2D mathematic features such as joints positions, angles, speed and angular velocity, etc., temporal-spatial features are 3D features based on 3D time and space of each joint. Because conventional motion features are 2D, a complete motion must be described by 2D motion features of all joints. But for 3D temporal-spatial features, each joint's features can represent a part of the whole motion independently. So we can process local temporal-spatial features of each joint separately and study the local similarity of each joint by decision tree to obtain their influence on global motion similarity.

Secondly, conventional motion features are extracted from original motion data, which have high time and space complexity with high dimension, so these methods need some dimension reduction algorithms. And 3D temporal-spatial features can avoid contacting with original motion data and eliminate "curse of dimensionality".

At last, after extraction of temporal-spatial features, 16 index lists for 16 human joints are built and similarity measure is implemented in each index list by DTW algorithm [7] separately. In order to learn the contribution of each joint during similarity measure, a data-driven decision tree is automatically constructed to reflect each joint's weight. Thus, we can measure similarity of joints with largest weight between motion example Q and motion A in the motion database: If these joints are dissimilar, motion A is skipped; if and only if similar, similarities of other joints between Q and A are calculated. It is obviously that this method can save computing time significantly.

2. Related Work

Until now, several motion features have been proposed: Y.Chui et al.[6, 8] proposed local spherical coordinates relative to the root orientation as the segments posture of each skeletal segment; Liu et al.[7] constructed a motion index tree based on a hierarchical motion description for motion retrieval; Lee et al.[4] described a two-layer structure for representing human Mocap data. The lower layer is a Markov process that retains the details of the original Mocap data, while the higher layer is a statistical model that provides support for the user interfaces by clustering the data to capture similarities among character states; Asaa et al[9] proposed a method of Action Synopsis, which used original motion capture data, such as joints positions, angles, speed, angular velocity, etc., to make some affinity matrices. Then a low dimension motion curve is produced from these affinity matrices by an RMDS method. The above-mentioned methods, which extracted features directly from original data, need some dimension reduction algorithms to avoid "Curse of dimensionality". Mueller et al.[10] introduced 31 Boolean features expressing geometric relations between certain body points of a pose. These geometric features seem to be a kind of local 3D semantic features. But they are extracted by the relationship of some neighbor joints, so geometric feature is a kind of 2D feature. And geometric feature is too complex, which is 31 dimensions.

For motion similarity measure, Y.Chui[6] proposed an index method by pose-based, then used DTW algorithm to calculate the actual distance between motions; Keogh[11] applied a kind of DTW algorithm based on bounding envelopes to speed up global similarity measuring; Kovar[12] queried similar motion automatically from a great deal of data and used querying result to build a continuous Parametric motion space. Then a new query example is extracted from a series of related motion.

3. 3D Temporal-spatial Features

By studying most kinds of motion data, space transformations of hands and feet are found to be regular. Different motions have different space transformations, such as hands and feet swing from back to front at the same speed back and forth, when a man is walking; hands moving from one side of body to another side, When a man is boxing, etc. So features with space transformations and time property can represent human motion clearly, we call them temporal-spatial features.

3.1. Motion Model

In this paper, a simplified human skeleton model is defined as Figure 1, which contains 16 joints that are constructed in the form of tree. Joint *root* is root of the tree and those paths from root to all endmost joints in human skeletal model from sub-trees of root.

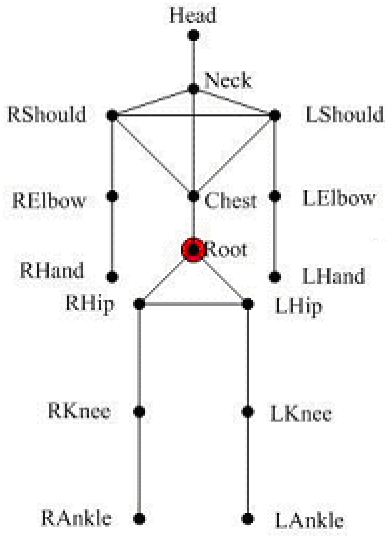

Figure 1. Human Skeleton

World coordinate of each joint can be represented as follow:

$$\vec{p}_i^{(j)} = T_i^{(root)} R_i^{(root)} \ldots T_0^{(grandparent)} R_i^{(grandparent)}(t) T_0^{(parent)} R_i^{(parent)} \vec{p}_0^{(j)} \quad (1)$$

where $\vec{p}_i^{(j)}$ is world coordinate of joint j at time i, $T_i^{(root)}, R_i^{(root)}$ are the position matrix and orientation matrix of root at time i, $T_0^{(k)}$ is position matrix of joint N_k (N_k is an arbitrary joint in the paths from Root to endmost joint in human skeleton tree) in local coordinate system of its parent at start time; $R_i^{(k)}$ is orientation matrix of joint N_k at time i, made by r_i^k, $\vec{p}_0^{(j)}$ is position matrix of joint N_j in local coordinate system of its parent at start time.

3.2. Temporal-Spatial Feature Extraction

According to Equation (1), we can calculate world coordinate of each joint and get 48 dimensional data.

Given a motion M consisting of n sampling frames, each motion can be represented as follow:

$$\left.\begin{aligned} M_s &= (F_1, F_2, \ldots, F_i, \ldots, F_n) \\ F_i &= (p_{i1}, p_{i2}, \ldots, p_{ij}, \ldots, p_{i16}) \\ p_{ij} &= (x, y, z) \end{aligned}\right\} \quad (2)$$

where n is the number of frames of motion data, p_{ij} is world coordinate of joint j at i th frame.

Now space transformations of each joint are calculated. Firstly, we define a space transformation set of upper body S_{up}, and a space transformation set of lower body S_{down} as following:

$S_{ui} \in S_{up}$, i=1,2…m; $S_{dj} \in S_{down}$, j=1,2…m; where m is the number of spaces in space transformation set, S_{up} and S_{down} have the same number of spaces. If we take Root as benchmark, then space transformations of joints above Root belong to S_{up}, and others belong to S_{down}, if a joint on upper body enters into space S_{ui}, its space transformation is S_{ui}.

Four space partition rules are defined as follow:

$$front(N_i, N_j) = \begin{cases} 1, N_i \text{ in front of } N_j \\ 0, N_i \text{ behind of } N_j \end{cases}$$

$$left(N_i, N_j) = \begin{cases} 1, N_i \text{ left to } N_j \\ 0, N_i \text{ right to } N_j \end{cases}$$

$$high(N_i, N_j) = \begin{cases} 1, N_i \text{ above } N_j \\ 0, N_i \text{ below } N_j \end{cases}$$

$$far(N_i, N_j) = \begin{cases} 1, N_i \text{ distance from } N_j > \lambda \\ 0, N_i \text{ distance from } N_j < \lambda \end{cases}$$

where rules of front, left and high depend on space relationship of up/down and left/right between joint N_i and N_j, rule of far depends on range of motion. As usual, in rules of front and left, N_j is Root, but in rules of high and far, N_j on upper and lower body are different. N_i, N_j are both at the same sampling frame.

Now we define motion space transformations:

B=$(S_1, S_2, ..., S_{16})'$, $S_i = (s_{i1}, s_{i2}, ..., s_{in})$.

where S_i is space transformation vector of joint i, n is the number of frames, s_{ip} is space transformation of joint i at p th frame. Suppose S_a is space transformation vector of joint a on lower body, S_a =($s_{a1}, s_{a2} \dots s_{aj} \dots s_{an}$):

Table 1. space rule table to calculate s_{aj}, N_{aj} is joint a at j th frame, N_{rj} is joint root at j th frame, N_{kj} is joint knee at j th frame.

S_{aj}	$front(N_{aj},N_{rj})$	$left(N_{aj},N_{rj})$	$high(N_{aj},N_{kj})$	$far(N_{aj},N_{kj})$
$S_{aj}=S_{d1}$	1	1	1	1
$S_{aj}=S_{d2}$	0	1	1	1
…	…	…	…	…
$S_{aj}=S_{dm}$	0	0	0	0

In table 1, some rules can be concluded:

If $s_{aj} = s_{d1}$ $\Leftrightarrow$ rule:

$$front(N_{aj}, N_{rj}) \wedge left(N_{aj}, N_{rj}) \wedge high(N_{aj}, N_{kj}) \wedge far(N_{aj}, N_{kj})$$

……

If $s_{aj} = s_{dm}$ $\Leftrightarrow$ rule:

$$\neg front(N_{aj}, N_{rj}) \wedge \neg left(N_{aj}, N_{rj}) \wedge \neg high(N_{aj}, N_{kj}) \wedge \neg far(N_{aj}, N_{kj})$$

The rules cited above are calculated by 48 dimensional data from Equation (2). Because these rules are all calculated at same frame, time and space complexity are not high. Moreover, space transformations of each joint are independent.

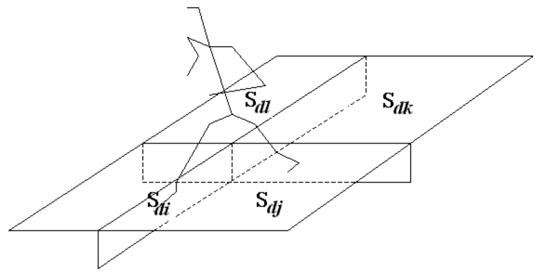

Figure 2. Space transformations of run's feet

For example, we extract local space transformations of motion run's (see figure 2) left foot and right foot as following:

$S_{leftfoot}$ =(S_{dk}, S_{dj}, S_{dk}, S_{dj}, …);

$S_{rightfoot}$ =(S_{di}, S_{dl}, S_{di}, S_{dl}, …);

Up to now, motion's space transformations are extracted, which is a kind of the reflection of motion spatial characteristic. But first of all, a complete motion is a group of time series data. Without time property, temporal-spatial features cannot represent motion clearly.

So the time property of motion is calculated as a part of temporal-spatial features.

The first time property is space transformation speed. Because each joint's space transformations are independent, space transformation speed is independent either. The algorithm can be summarized as follow:

Procedure SpaceSpeed()

Input: local space transformation vector of k^{th} joint

$S_k = (S_{k1}, S_{k2}, \ldots, S_{kn})$,n is the number of frames.

Output: $SP_k = (SP_{k1}, \ldots, SP_{ki}, \ldots)$,SP_{ki} is space transformation S_{ki}'s speed of k^{th} joint.

(1) Initialization: $num_j = 0$,i=1,j=0,L=S_{ki}

(2) if $S_{ki} \neq S_{k(i+1)}$, {$spacespeed_{kl} = num_j$,l=$S_{k(i+1)}$,j=j+1}

else $num_j = num_j + 1$;

(3) i=i+1,if meet the end of frames goto (4) else goto (2)

(4) return SP_k

This spacespeed is actually the speed of a joint moving from a space to another. The weighted sum of every joints' spacespeeds consists of the whole motion's spacespeed.

During similarity measure, because of irregularity and contingency of human motion, there are odd space transformations that cannot be matched. Therefore spacenoise is defined to measure some odd space transformations.

Procedure SpaceNoise()

Input: local space transformation vector of k^{th} joint

$S_k = (S_{k1}, S_{k2}, \ldots, S_{kn})$,n is the number of frames

Output: $SpaceNoise_k$

(1) Initialization: $num_j = 0$,i=1,j=0,l=1

(2) if $S_{ki} \neq S_{k(i+1)}$

Noise= num_j , j=j+1,

if $\frac{Noise}{n} < \varepsilon$ add S_{ki} to $SpaceNoise_k$

else

$num_j = num_j + 1$;

(3) i=i+1,if meet the end of frames goto (4) else goto (2)

(4) return $SpaceNoise_k$

As space transformations, spacespeeds and spacenoises of 16 joints are gotten, complete temporal-spatial features are formed through the merger of them.

3.3. Keyspace Construction and Indexing

Since space is continuous, a joint is always in same space at some consecutive frames, which is spatial invariant. Therefore frames with the same temporal-spatial features can be merged into a keyspace. The keyspace is similar to the keyframe, but the keyspace is the local concept for each joint and keyframe is the global one for the whole motion. The following algorithm is proposed to calculate keyspace:

Procedure keyspace()

Input: local temporal-spatial features of k^{th} joint

$st_k = (st_{k1}, st_{k2}, \ldots, st_{kn})$. n is the number of frames

Output: $keyspace_k = (\ldots, st_{ki}, st_{k(i+1)}, st_{k(i+2)} \ldots)$

$(st_{ki} \neq st_{k(i+1)}, st_{k(i+1)} \neq st_{k(i+2)})$

(1) Initialization: i=0, $S_{ki} = \phi$

(2) i=i+1,if meet the end of frames,goto (4) else goto (3)

(3) if $st_{ki} \neq st_{k(i-1)}$ {add st_{ki} to $keyspace_k$,goto (2)}

else goto (2)

(4) return $keyspace_k$,S_p

Now the space transformation is simplified to the keyspace of each joint and similarity measure can implement in keyspaces directly that can reduce time and space complexity.

We build a motion retrieval database with 16 index lists by temporal-spatial features of each joint. For a query, the similarity measure is for each joint and so it is the local measure. In this study, it can apply local DTW algorithm [7].

DTW algorithm is a nonlinear match method originally used in speech recognition and it has been successfully applied to motion capture data processing.

Given two motions, M1 {$S1_1$, $S1_2 \ldots S1_N$}, M2 {$S2_1$, $S2_2 \ldots S2_M$} are temporal-spatial features of k^{th} joint. The distance between them is defined as follow

$$D = \frac{1}{2}\left(\min_{\{\omega 1(i)\}} \sum_{i=1}^{N} d(i, \omega 1(i)) + \min_{\{\omega 2(i)\}} \sum_{i=1}^{M} d(i, \omega 2(i))\right)$$

$d(i,j)$ is the distance between the i^{th} temporal-spatial features of M1 and the j^{th} temporal-spatial features of M2,defined as:

$$d(i,j) = \sqrt{\sum_k w_k (S1_{ik} - S\mathbf{2}_{jk})^2}$$

where $S1_{ik}$ is i th temporal-spatial features of M1, $S2_{jk}$ is j th temporal-spatial features of M2, w_k is the weight.

Let the left half part of D be DA, defined as

$$DA = \min_{\{\omega 1(i)\}} \sum_{i=1}^{N} d(i, \omega 1(i))$$

We solve it recursively by applying dynamic programming in following way:

The time warping path ω1(i) is constrained by the following boundary and continuity conditions.[13]

1) The boundary conditions ensure that the first and last temporal-spatial features of M1 are matched to the frame b and frame e of M2 ω1(1)=b, ω1(N)=e:

$$b = \min_{i \le M} \arg(d(1,i) \le threshould)$$

$$e = \max_{i \le M} \arg(d(N,i) \le threshould)$$

2) The continuity conditions restrict the match of the intermediate frames, and ω1(i) is defined as a monotonically increasing function :

$$DA(i,j) = d(i,j) + \min\{DA(i-1,j), DA(i-1,j-1), DA(i-1,j-2)\}$$
$$DA(1,b) = d(1,b)$$

where $DA(i$-1,j-2) corresponds to skipping the (j-1) th temporal-spatial features of M2, and $DA(i$-1,$j)$ means that at least two temporal-spatial features of M1 correspond to j th temporal-spatial features of M2.

Since DTW is cost-intensive in computing time and memory, similarity measure by DTW for temporal-spatial features of all joints still has high time and space complexity. We will discuss how to resolve it in Section 4.

3.4. Semantic Features

Temporal-spatial features can be used to extract semantic features of common motions.

For running (see figure 2), man's left foot is moving from S_{dk} to S_{dj}, then moving from S_{dj} to S_{dk}, and so on. And the movement of right foot is between S_{di} and S_{dl}. Furthermore, moving speeds of two feet are almost the same, so are moving ranges. It is widely known that a man's movement of two hands is similarity to his feet when he is running or walking. So we can summarize the rules of temporal-spatial features for some important joints of running. These rules are semantic features. As mentioned above, most kinds of motions have their own semantic features.

By semantic features, some common motions are recognized and retrieved automatically, an inverse retrieval system is made by using keyword (e.g. walk, jump) instead of query example to implement motion retrieval and save retrieval time. Now our system can extract semantic features of some simple motions (such as walk, run, wave, jump) by temporal-spatial features.

4. Decision Tree

In sect.3.2, DTW algorithm with temporal-spatial features of all joints has high time and space complexity. In the study, we find that different joints of human have different effects on motion similarity measure. For most motions, two hands and two feet have the greatest effect. After learning a great deal of data, a rule is found that if temporal-spatial features of hands and feet are the same, it need temporal-spatial features of some other joints to measure similarity of two motions, but if two motions are dissimilar, their temporal-spatial features of hands and feet are always dissimilar too. So each joint's effect on the global motion is meaningful to motion retrieval

Since the intuition is inaccuracy, judging the joints' effect on human motion cannot simply rely on our intuition .So the method based on data driven decision tree learning is applied for training human motion data to build a decision tree with importance of each joint. The similarity measure can be implemented in order of joints importance to speed up motion retrieval by this decision tree. In this paper, decision tree learning is not referred to spacespeed and spacenoise.

4.1. Decision Tree Induction

The earliest system of decision tree is CLS designed by Hunt[14].This method with the help of the intuition is short of mathematics foundation. Now the methods based on probabilistic theory are universally adopted. One of the well-known methods is a decision tree induction algorithm: ID3, proposed by Quilan in 1986[15]. After ID3, Quilan also gave some improved algorithm, such as C4.5 [16], C5.

The principle of ID3 is that the choice of attribute should maximize the information gain and minimize the Entropy while the average test times from root to leaf node is minimal in probabilistic.

Given some classes, the number of classes is c, if attribute A=v in all examples, its Entropy is defined to be:

$$\text{Entropy (A=v)} = \sum_{i=1}^{c} - p_i \log_2 p_i$$

The information gain of attribute A is defined as the difference between the original Entropy and the new Entropy after classifying train examples by attribute A.

$$\text{Gain(T,A)=E(T)} - \sum_{j=1}^{V} \frac{|T_j|}{|T|} E(T_j)$$

where T is train set of examples, T_j is a train set of examples when A=j, $T_j \subset T$

The ID3 algorithm can be summarized as follow:

Procedure ID3()
Input R: unclassific attribute, C: classific attribute, T: train set
Output: decision tree Tr
(1) if T is empty , return a node which value is 0;
(2) if all train examples in T are in same class, namely same attribute value, return a node with this attribute value.
(3) if R is empty , return a node with the most frequently attribute value.
(4) D= maximum(Gain(D,T)) in R
(5) $\{d_j \mid j=1,2\ldots m\}$ = attribute D's value

$\{S_j \mid j=1,2\ldots m\}$ is subset of S, the attribute D's value of all examples in S_j is d_j

(6) return a tree, rood node is marked D, paths from root is marked $d_1, d_2, \ldots, d_m$, paths correspond to subtree respectively:
ID3(R-{D},C,S_1),
ID3(R-{D},C,S_2),…,ID3(R-{D}, C, S_m);

A decision tree is constructed automatically by training six hundred human motions in database. (see figure3).

Now can conclude that two feet's temporal-spatial features have the greatest effect on human motion similarity, and two hands are the second. And the influences of other joints cannot be compared with. Given a query example A and a motion B in the database, during the similarity measure, two feet's temporal-spatial features should be calculated firstly; if they are similar, two hands are calculated secondly; if dissimilar, Motion B is skipped and a new motion is taken from database for motion retrieval. Other joints' temporal-spatial features are calculated only if temporal-spatial features of two feet and two hands are similar. Consequently, Retrieval based on decision tree learning avoids a great deal of meaningless DTW calculating and becomes more efficient.

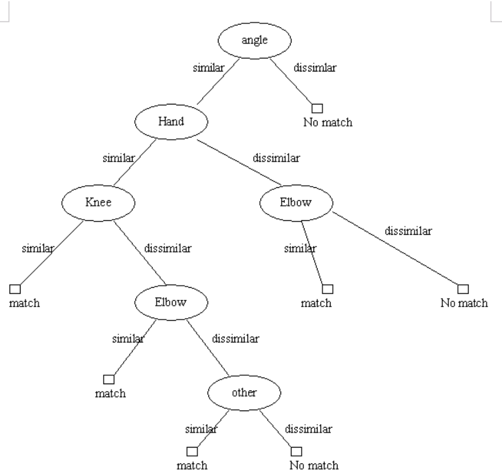

Figure 3. Decision tree during motion retrieval

4.2. Performance of Decision Tree

Now a methodology for assessing prediction quality after the fact is introduced [17]:

(1) Collect a large set of example

(2) Divide it into two disjoint sets: the training set and the test set.

(3) Use the learning algorithm with the training set as examples to generate a hypothesis H.

(4) Measure the percentage of examples in the test set that are correctly classified by H

(5) Repeat steps 1 to 4 for different sizes of training sets and different randomly selected training sets of each size

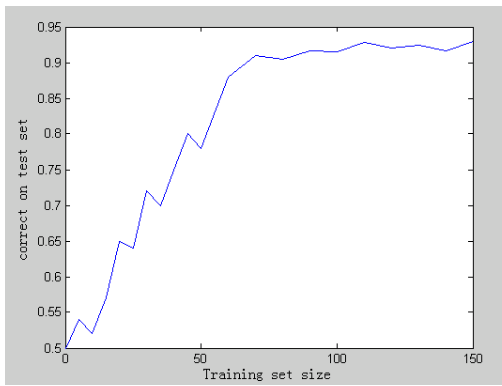

Figure 4. The learning curve of decision tree built by motion capture data.

The learning curve for the decision tree with motion capture data is shown in figure4.

Notice that as the train set exceeds 70, the prediction quality remains stable. So the algorithm of data driven decision tree is reliable.

5. Experiment Results

We implement our retrieval algorithm in matlab. It is more than 1300 real human motion captured sequences with different types in motion database for test. Different types of query examples and key words of some common motions can be used. We also summarize some semantic features by temporal-spatial features to recognize some types of motions.

Table 2. A(43) is a motion A with 43 frames, N is the number of motion in database

Motion sequences	Query time			
	Similarity measure of all joints		Based on data driven decision tree	
	N=200	N=800	N=200	N=800
A(43)	1.5470s	4.7817s	0.7851s	1.9112s
B(90)	1.7900s	5.1021s	0.8191s	2.0121s
C(150)	1.9145s	5.5121s	0.8823s	2.0376s
D(240)	2.1219s	5.6546s	1.0656s	2.1578s

Table 2 shows that the retrieval time of our method by decision tree is so much less than time of conventional method that is measuring all human joints with different weight that the retrieval performance has been improved significantly. And the querying time to process a query very much depends on the size of database.

Table 3: KF is the method of motion retrieval based on bone feature and keyframe, SF&DT is the method of motion retrieval based on temporal-spatial features and data driven decision tree.

Motions	Recall		Precision	
sequences	KF	SF&DT	KF	SF&DT
Walk	0.85	0.89	0.90	0.95
Run	0.70	0.85	0.85	0.95
Punch	0.40	0.75	0.60	0.85
kick	0.50	0.75	0.70	0.85

To compare motion retrieval efficiency of the proposed method with that of the method based on bone angles feature and keyframe extraction, table 3 shows the recall and precision of these two methods in the same motion database. It is obvious that the retrieval accuracy of the proposed method is better than the other one.

6. Conclusion and Future Work

In this paper, temporal-spatial features are proposed which describe 3D space relationship of each joint and each joint has its own independent temporal-spatial features. If consecutive frames yield the same temporal-spatial features vectors, we refer to them as keyspaces. The index lists of 16 joints are then built by their keyspaces. The data-driven decision tree is constructed automatically and the joint in higher level in this tree means greater influence on global motion match. During motion retrieval, more important joints have higher priority on similarity measure. At last motion retrieval is sped up significantly. Semantic features of some common motions can be extracted from motion temporal-spatial features.

In future, we will address three questions in the following: (1) The performance of temporal-spatial features largely depends on space partition. We will find some new methods of space partition for more motions. (2) The statistics method can be used to analyze semantic features and annotate new motion capture data automatically. (3) Based on data driven decision tree learning, the method of multiple-instance learning will be used to improve the accuracy and availability of decision tree.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No.60533090, No.60525108), 973 Program (No.2002CB312101), Science and Technology Project of Zhejiang Province (2005C13032, 2005C11001-05).

References

[1] Yueting Zhuang, Yong Rui, Thomas S.Huang. Adaptive Key Frame Extraction Using Unsupervised Clustering[C]. Proceedings of IEEE Int'l Conference on Image Processing, Chicago, USA. 1998.

[2] W.Wolf. Key frame selection by motion analysis[C] Proceedings of the IEEE International Conference on acoustics, Speech, and Signal Processing(ICASSP), atlanta, USA, 1996,Vol 2, pp1228-1231.

[3] K.S.Beyer, J.Goldstein, R.Ramakrishnan, and U.Shaft. "When is 'nearest neighbor' meaningful?"[C], Proceedings 7th International Conference on Database Theory(ICDT'99), Jerusalem, Israel, 1999, pp.217-235.

[4] I.S.Lim, D.Thalmann, Key-Posture Extraction Out of

Human Motion Data by Curve Simplification[C], 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2001, vol 2, 1167-1169.

[5] Jehee Lee, Jinxiang Chai, Paul S.A.Reitsma, Jessica K.Hodgins and Nancy S.Pollard. Interactive Control of Avatars Animated with Human Motion Data[C]. In proceedings of SIGGRAPH 2002, San Antonio, Texas, 2002, 491-500.

[6] C.Y.Chui, S.P.Chao, M.Y.Wu, S.N.Yang, and H.C.Lin. Content-based Retrieval for Human Motion Data[J]. Journal of Visual Communication and Image Representation, 2004, Vol 16, Issue 3, pp 446-466,

[7] Liu F., Zhuang Y., Wu F., and Pan Y.H, 3d Motion Retrieval with Motion Index Tree[J]. Computer Vision and Image Understanding, 2003, 92(2-3): 265-284.

[8] Shi-pin Chao and Chih-Yi Chiu, Motion retrieval and synthesis based on posture features indexing[C]. The 5th International Conference on Computational Intelligence and Multimedia Applications,Xi'an china, 2003.

[9] Assa J, Caspi Y, and Cohen-Or.D. 2005. Action synopsis: pose selection and illustration[J]. ACM Trans. Graph, 2005, 24(3): 667-676.

[10] Meinard Mueller, Tido Roeder, Michael Clausen. Efficient Content-Based Retrieval of Motion Capture Data[C]. Proceedings of ACM SIGGRAPH 2005, 2005,24(3): 677-685.

[11] Keoph.E, Palpanas, T.Zordan V.B, Indexing large Human-motion database[C]. Proceedings of the 30th. VLDB Conference, Toronto, Canada, 2004, pp. 780-791.

[12] L.Kovar, M.Gleicher. Automated Extraction and Parameterization of Motions in Large Data Sets[J]. ACM Trans. Graphics, 2004, 23(3): 559-568.

[13] C.C.Tappert, Adaptive on-line handwriting recognition [C]. In Proceeding of International Conference on Pattern Recognition, Montreal, Canada, 1984, 1004-1007.

[14] E.B.Hunt, J.Marin, and P.T.Stone. Experiments in induction[M]. Academic Press, New York, 1966.

[15] Quilan JR. Generating Production rules from decision tree[C], Proceedings of the 10th International Joint Conference on Artificial Intelligence, 1987, pp.304-307.

[16] Quilan JR. C4.5: Programs for machine Learning[C], San Mateo, CA: Morgan Kaufmann,1993

[17] Stuart Russell and Peter Norvig. Arfificial Intelligence: A Modern Approach Second Edition[M], Prentice Hall, Englewood Cliffs, New Jersey, 2002.