

## SPECIAL ISSUE PAPER

# Human motion retrieval based on freehand sketch

Zhangpeng Tang<sup>1</sup>, Jun Xiao<sup>1\*</sup>, Yinfu Feng<sup>1</sup>, Xiaosong Yang<sup>2</sup> and Jian Zhang<sup>2</sup><sup>1</sup> Institute of Artificial Intelligence, Zhejiang University, Hangzhou, Zhejiang, China<sup>2</sup> The Media School, Bournemouth University, Poole, Dorset BH12 5BBUK

## ABSTRACT

In this paper, we present an integrated framework of human motion retrieval based on freehand sketch. With some simple rules, the user can acquire a desired motion by sketching several key postures. To retrieve efficiently and accurately by sketch, the 3D postures are projected onto several 2D planes. The limb direction feature is proposed to represent the input sketch and the projected postures. Furthermore, a novel index structure based on k-d tree is constructed to index the motions in the database, which speeds up the retrieval process. With our posture-by-posture retrieval algorithm, a continuous motion can be got directly or generated by using a pre-computed graph structure. What's more, our system provides an intuitive user interface. The experimental results demonstrate the effectiveness of our method. Copyright © 2014 John Wiley & Sons, Ltd.

## KEYWORDS

motion retrieval, sketch, motion index, motion graph, computer animation

### \*Correspondence

Jun Xiao, Institute of Artificial Intelligence, Zhejiang University, Hangzhou, Zhejiang China.

E-mail: junx@cs.zju.edu.cn

## 1. INTRODUCTION

With the development of motion capture techniques, motion capture data has been widely used for capturing character motions in many applications such as performance animation, video games, featured films, and virtual reality [1,2]. The manipulation of motion capture data such as motion retargeting [3,4], motion synthesis [5–8], and motion editing [9,10] needs an easy, efficient, and accurate method to retrieve similar motions from the large amount of recorded motion capture data.

To get a desired motion from large human motion repository, the input of the retrieval system is a very important consideration for the animators. Prior works usually utilize the motion textual [11] such as 'running', 'kicking' and 'fighting', or using a sample motion clip [12–14] as the input of retrieval system. Motion textual is efficient to find similar motions, but annotating large amount of motions in the repository requires plenty of manual work and textual cannot always describe human motion sufficiently. For example, it is hard to describe a complicated motion such as 'run in the first half, several loops in the second half', and it is difficult to decide on a suitable textual for two different free style dances. Using motion clip as the query term is easy to measure the similarity between clips, especially the similarity between single postures, but the query clip is usually hard to acquire, the

user may have to examine a large number of examples in the motion repository by reviewing the motions. Although N.Naoki *et al.* [15] use a puppet interface to generate query motion clip, the puppet instrumented with sensors and potentiometers is not an easy-obtain device. In this paper, the hand-draw sketch is utilized as the input. Sketching interface is intuitive even for novices, because most people grew up scrawling and drawing some figure line. Some simple constraints such as the first stroke that must be torso are appended in our sketching interface.

The efficiency and accuracy are the key issues of a retrieval system. What the user cares about is whether the retrieval result is similar with what they want and the retrieval efficiency. The major challenge for the sketching retrieval system is matching the given 2D sketches to the appropriate sequences in the 3D motion database. In this work, we extract the key postures of the 3D motions and then project them into several 2D planes. After that, a great amount of projected postures can be obtained. The limb direction feature is proposed, which can match sketches with projected postures accurately. A novel method based on k-d tree [16] is used to index the LD features of projected postures, which is efficient for searching large amount of human motions. Sometimes, there is not a desired motion in the database for the user, but it can be synthesized by blending some existing motion segments

from different motions. Here, motion graph based on Kovar's work [7] is used to generate continuous motion.

An intuitive user interface makes the retrieval system easy to use for both the professional animators and the ordinary users. We provide an interface on which the user can draw any number of sketching key postures. Only some very simple rules have to be obeyed. The sketching postures can be modified at any time. For the sketching posture, our system provide a group of candidate postures so that the user can select several best matches.

This paper is organized as follows. We first cover related work in Section 2. Section 3 gives an overview of our human motion retrieval framework. Section 4 describes the sketching interface of our system. The approach of motion index based on k-d tree and the LD feature is introduced in Section 5. Section 6 presents the retrieval algorithm and how to build motion transition graph. Finally, we present the experimental result and conclusion in Section 7 and Section 8.

## 2. RELATED WORK

Motion retrieval is a key issue of motion synthesis and motion reuse, which have been studied for over a decade. The existing motion retrieval approaches can be classified into three categories based on the query input item: motion clip, puppet, and sketch.

Because of the complexity of motion data, the motion clip itself would be the best query to search for similar motions. For instance, Chiu *et al.* [17] propose a two-stage motion retrieval system. They use an index map structure based on an affine invariant posture feature to find the candidate clips, then Dynamic Time Warping (DTW) is used to calculate the similarity between query example and each candidate clip. However, the high computational complexity of DTW make it difficult to meet the real-time motion retrieval requirement. Müller *et al.* [14] use boolean geometric posture feature that describes geometric relations between several specified body joints for an efficient comparison between different motion data. Deng *et al.* [18] suggest a motion pattern discovery and matching approach that breaks human motions into a part-based, hierarchical motion representation. They use a fast string matching algorithm for efficient runtime query processing. Zhu *et al.* [19] present a quaternion space decomposition method, they use quaternion combination to represent the high dimensional human motion data. All methods mentioned above require a query motion clip, which is similar to the desired one. When the user does not have a similar clip, too much effort must be taken capturing the query clip, and some motions are hard to perform such as back flip and handstand walk.

A few researchers utilize instrumented puppet as the input interface. Johnson *et al.* [20] develop an instrumented puppet to control a bird-like character, and the Hidden Markov Models (HMMs) is used to recognize the user's manipulation. Feng *et al.* [21] acquire the query key postures by posing a wooden artist's doll with painted joints

in front of a stereo camera. Numaguchi *et al.* [15] present a puppet instrumented with orientation sensors and potentiometers. The user can control the puppet to performing an approximation of a desired motion. However, the cost for devices and the operational complexity are the limitations of these methods.

In recent years, sketch input was gradually used in character animation. Thorne *et al.* [22] introduce an approach where animated motion is created for the character by drawing a continuous sequence of lines, arcs, and loops. Then they are mapped to a parameterized set of output motions. Li *et al.* [23] propose a sketch-based approach for creating Kung-fu motions. They retrieve similar motions by sketching the initial and closing posture of a Kung-fu motion and the trajectories on specific moving joints. But their sketch is a kind of complex robot style, which is difficult to draw for unprofessional users and the joint location must be labeled manually. Choi *et al.* [24] suggest a retrieval method by hand-drawn stick figure, they convert the 3D motions into 2D stick figure style on a selected plane. Then, the user retrieves the motions by drawing some stick figure style sketches after reviewing the generated stick figures database. But the view plane selected by the user may not be the same as what their algorithm select, and their retriever algorithm, which will search the whole database, spends too much time when the database is very large. In our work, the 3D posture are projected into multiple planes, and the motion index helps speeding up the retrieval process. What is more, some researches [25–27] use sketch to define the trajectory constraints that the character or its specific joint move along. However, the user still has to provide some 3D query postures.

## 3. OVERVIEW

Our retrieval method can be divided into two stages: the preprocessing stage and the run-time stage. Figure 1 shows the framework of our method. In the preprocessing stage, we have two assignments. First, we construct a k-d tree index for the whole motion database based on key posture extraction and LD feature. Then, we build a motion graph by clustering the postures of all the motions.

In the run-time stage, the user inputs several sketching postures, then the joints are labeled automatically and the limb feature is extracted for each posture. Next, we apply the posture-by-posture retrieval algorithm. If a result motion is returned directly, which means a motion in the database is the user's target, the retrieval is over. Otherwise, the system will travel the motion graph to find a shortest transition path and generate a new consecutive motion.

## 4. SKETCHING INTERFACE

The sketching interface allows the user to define a desired motion by drawing several simple motion strokes. Before using our system, the user is required to obey some rules.

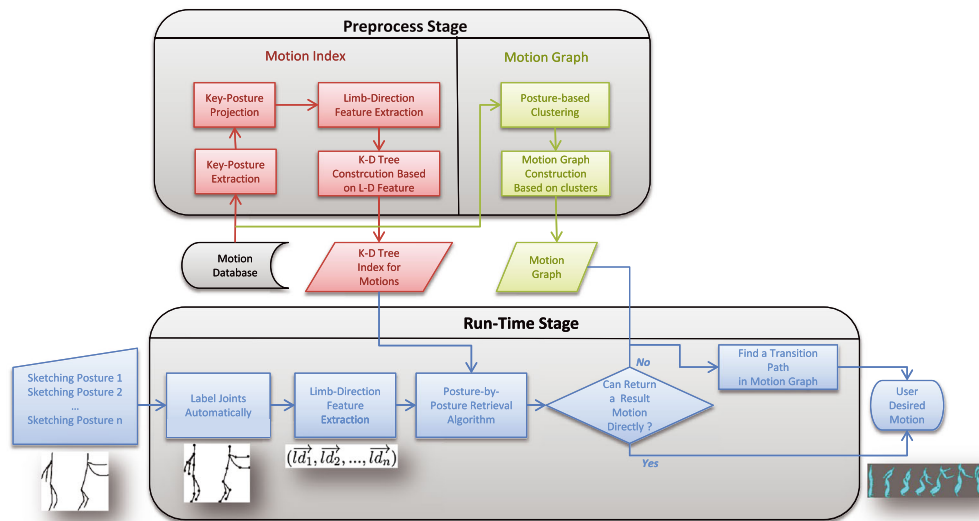


Figure 1. The framework of our motion retrieval method.

These rules are very easy to learn, which will never be obstacles for the novices. When the user finishes a posture sketch, the joints are labeled automatically.

#### 4.1. User Rules

**View plane:** Generally speaking, people would draw some key postures from a specific view plane when they want to express a certain motion. And the view planes selected by different users are not always the same. For a 3D motion, its projection is quite different when the view plane is changed. Generally, the view tilt and elevation angles are set to zero and a azimuth angle that makes the motion easily distinguished is selected. For example, when viewing a running motion, the side view is a better choice because it describes the moving direction clearly. For a waving hand motion, the front view will be chosen mostly.

**Key postures:** Drawing some key postures is an effective and succinct way to express a desired motion. The user is required to select the postures where the motion changes the direction or stops [24]. For instance, The key postures of the boxing motion are at the moment where the fist reach out and the fist take back. Besides, the user is required to select a representative posture as the first sketch, which is beneficial to our retrieval algorithm.

**Sketching order:** In our system, a sketching posture is limited to five parts that include torso, two upper limbs, and two lower limbs. The head is included in torso. In order to make the automatic joints labeling easier, the user is required to sketch a posture by starting with drawing the torso. And then right upper limb, left upper limb, right lower limb and left lower limb are drawn orderly.

**Posture rotation:** In our retrieval system, we consider them to be different postures if a man stand or lie with the same pose, which should be obeyed when the user draws the input postures.

These rules are very simple for a professional animator. They are also not an obstacle for the ordinary user after some training.

#### 4.2. Automatical Joints Labeling

In this paper, the joints considered are head, neck, elbows, hands, spine, pelvis, knees, and feet (12 in total). Figure 2 shows an example of sketching posture. The joints are labeled automatically along with user's input. Besides, Spline interpolation and smoothness are applied to the

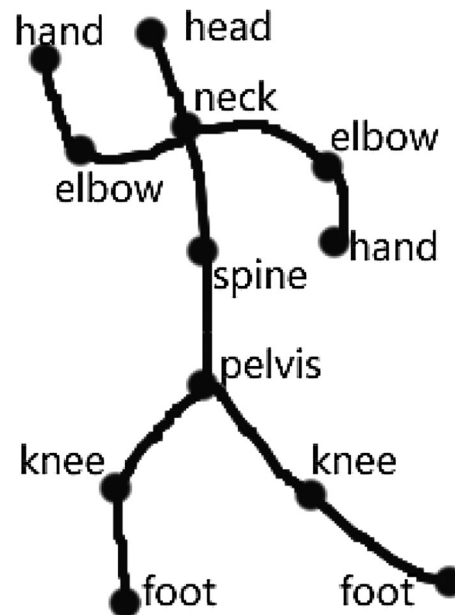


Figure 2. A freehand sketching posture. The joints are labeled automatically. The stroke between two adjacent joints is called limb in this paper.

user's strokes. The first stroke is regarded as torso. The two endpoints of torso are labeled as head and pelvis, respectively. When two upper limbs are finished, the intersection of torso, and the straight line connecting the starting points of two upper limbs is labeled as neck. Obviously, the two end points of upper are hands. The middle point of neck and pelvis is the spine. Next, two endpoints of lower limbs away from pelvis are labeled as feet. For the high flexibility of the elbow and knee and the different style of the user's sketch, we cannot easily label the middle point of limbs as elbow or knee just as Choi's work [24]. Here, we detect the corner point of the limb and label it as elbow or knee. If no corner point is detected, the middle point is selected.

## 5. MOTION DATA INDEX

There are masses of motions in a human motion database. So comparing the input sketch with projected-postures one by one is time-consuming, which does not meet the efficiency of some applications. In this paper, we propose an effective method to index the motions in database. It contains four procedures in total.

### 5.1. Key Posture Extraction

An original motion in database includes large amount of postures. The adjacent postures in a motion is always similar with each other. So some key postures can be selected to represent the motion just as the users express their desired motion by sketching several key postures.

Qi *et al.* [13] propose an approach based on  $k$ -means cluster, the motion postures are clustered into several classes, and all the class center postures are selected as the key postures. But the number of classes in  $k$ -means must be manually set in their method. Peng *et al.* [28] introduce another cluster-based method for motion data preprocess. Inspired by their work, a cluster-based and fully automatic key posture extraction approach is proposed in this paper.

At the beginning of clustering for a motion, we initialize a new cluster and set the first posture to be its center posture. For the incoming posture later, the distance between the new posture and the center posture of last cluster is computed. If the distance is greater than a user-specified threshold, a new cluster is created and current posture is specified to be the center posture. If the distance is less than threshold, we continue to the next posture. Finally,

the center postures of all the clusters are selected as key postures. Two examples presented in Figure 3 show the validity of our method. The distance function refers to Lee's [25]. We remove the spacial distance terms of two postures in their formula, which is used to calculate the cost for transiting a posture to another, because only the geometric similarity is considered in our work and the spacial position has no effect on our posture similarity. It is computed as

$$d(p_i, p_j) = \sum_{k=1}^n \omega_k \|\log(q_{j,k}^{-1} \cdot q_{i,k})\|^2 \quad (1)$$

where  $q_{i,k} \in \mathbb{S}^3$  is the orientation of joint  $k$  with respect to its parent in posture  $i$ , and joint angle differences are summed over  $n$  rotational joints.  $\log(q_a^{-1} \cdot q_b)$  is a vector  $V$  such that a rotation of  $2\|V\|$  about the axis  $\frac{V}{\|V\|}$  takes a body from orientation  $q_a$  to orientation  $q_b$ . Weights  $\omega_k$  is set to one for the 12 selected joints, and it is set to zero for the rest joints such as ankle, toes, and wrists, which have less effect on obvious differences between two postures.

### 5.2. Posture Projection

To compare the similarity of 3D posture and 2D sketching posture, the 3D posture is projected to 2D plane. Prior work [24] tries to find a best projection plane that maximizes the standard deviation of the projected 2D points, that makes the widest image of the 2D projected posture. But with some motions such as walk or run, the user tends to sketch them on side view, which may not make the widest image of 2D posture. In this paper, we set the view directions parallel to the  $XOZ$  plane and apply orthographic projection. For each key posture, eight projections are calculated with respect to view directions  $\{\vec{D}_k | k = 1, 2, \dots, 8\}$ , where

$$D_k = \left( \sin\left(\frac{k-1}{8} \cdot 2\pi\right), 0, \cos\left(\frac{k-1}{8} \cdot 2\pi\right) \right)$$

Figure 4 shows an example of posture projection.

### 5.3. Limb Direction Feature of Posture

In the process of retrieval, the features are extracted from input sketch and the projected posture, then the distance between them is computed. The feature has to describe the

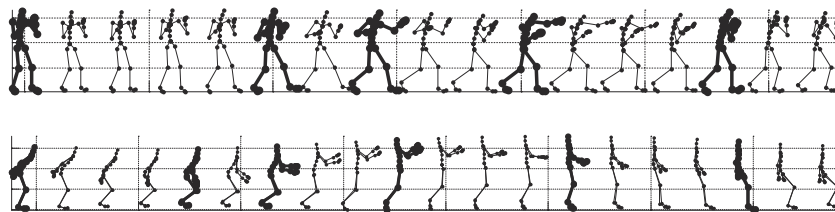
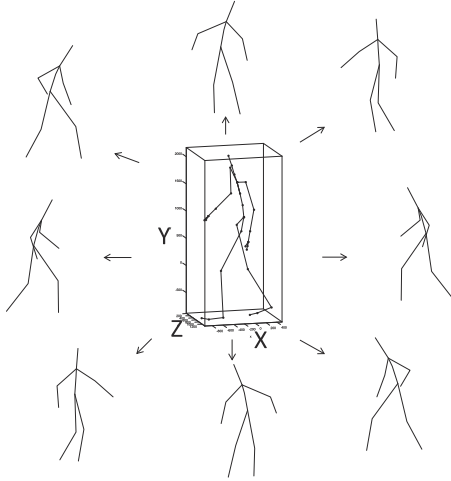


Figure 3. Two examples of key posture extraction. The thick ones are key postures.



**Figure 4.** The 3D posture is projected onto eight directions. The project directions are parallel to the XOZ plane.

similarity between the user's sketch and the projected posture accurately; meanwhile, it cannot be affected by the user's changeful input. Each user has different sketch style, the body proportion is hard to be unified for the same posture. So some information such as limb (as illustrated in Figure 2, limb is defined as the part of body that links two joints) length can not be used as an effective feature for the posture similarity comparison. In this paper, the direction of limb is utilized as 2D posture feature. Limb direction can express posture intuitively and accurately, and it can be computed efficiently. The LD feature is calculated as

$$LD = (\vec{ld}_1, \vec{ld}_2, \dots, \vec{ld}_n) \quad (2)$$

where

$$\vec{ld}_i = \left( \frac{x_i}{\sqrt{x_i^2 + y_i^2}}, \frac{y_i}{\sqrt{x_i^2 + y_i^2}} \right), \quad (3)$$

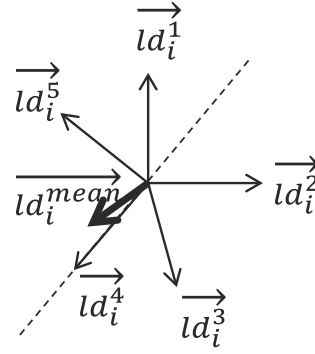
$$x_i = p_i^x - q_i^x, y_i = p_i^y - q_i^y$$

where  $\vec{ld}_i$  is the normalized direction vector corresponding to the  $i$ -th limb.  $n$  is the number of limbs.  $x_i$  and  $y_i$  are the elements of  $\vec{ld}_i$ ,  $p_i$  and  $q_i$  are the joints at the endpoints of the  $i$ -th limb,  $(p_i^x, p_i^y)$ , and  $(q_i^x, q_i^y)$  are their 2D coordinates.

Limb direction feature is not rotation invariant, which meets the last rule we make for the user in Section 4.1

#### 5.4. Feature Index with K-D Tree

In the motion database, there are many similar segments between different motions. For example, the high jump motion and the broad jump motion have the same running segment that appears in their first half. So, after key posture extraction for all the motions in database, a lot of key postures can be acquired, in which we can find many similar postures. The feature vectors of these similar key postures



**Figure 5.** The selection of partition node in k-d tree construction. The  $i$ -th limb in the fourth posture is chosen as the partition node.

lie closely in the feature space. For efficient retrieval, we utilize k-d tree to construct a motion index.

Traditionally, one dimension of the feature vector is selected to partition the feature space in each iteration of k-d tree construction. In this paper, a LD vector  $\vec{ld}_i$  that is two dimensions is chosen as the partition basis. The limb whose directions makes the maximal variance is chosen as partition basis in each iteration. The variance of the  $i$ -th LDs is calculated as

$$\sigma_i = \frac{1}{n} \sum_{k=1}^n \left[ 1 - \cos(\vec{ld}_i^k, \vec{ld}_i^{mean}) \right]^2 \quad (4)$$

where  $n$  is the number of posture in current iteration of k-d tree construction,  $\vec{ld}_i^{mean} = \vec{ld}_i^1 + \vec{ld}_i^2 + \dots + \vec{ld}_i^n$  and  $\vec{ld}_i^k$  is the direction vector corresponding to the  $i$ -th limb in the  $k$ -th posture, which is calculated as Formula 3.  $\cos(\vec{a}, \vec{b})$  is the cosine distance between two vectors. Assume the  $i$ -th limb is chosen as the partition basis in current iteration and  $n = 5$ . As shown in Figure 5, the  $i$ -th limb in fourth posture, which is closest to  $\vec{ld}_i^{mean}$ , is chosen as the current partition node. The searching space is divided into two subspaces equally by the dotted line.

Because of the deviation of hand-drawn sketch, the LD of sketch and projected-posture can not be matched accurately. So some modifications is made to the k-d tree searching algorithm. Assume that we arrive at the partition node in Figure 5 when searching the k-d tree. If  $1 - \left| \cos(\vec{ld}_i^{target}, \vec{ld}_i^4) \right| < threshold$ , which means the direction of the  $i$ -th limb in the target posture is close to the dotted line in Figure 5, both the left and right space of the partition node should be searched, otherwise the space that  $\vec{ld}_i^{target}$  lies in is searched.

## 6. RETRIEVAL

Given an input sketching posture  $S$  and a projected 2D posture  $P$ . The similarity between them is calculated as follows:

$$Sim(S, P) = \frac{\sum_{i=1}^n \cos(\theta_i)}{n} \quad (5)$$

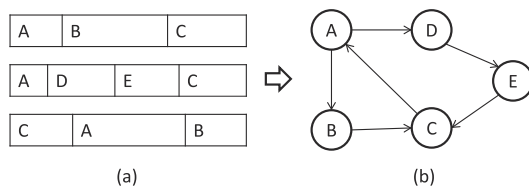
where  $\cos(\theta_i) = \frac{\vec{ld_i^S} \cdot \vec{ld_i^P}}{\|\vec{ld_i^S}\| \|\vec{ld_i^P}\|}$  is the cosine distance between the  $i$ -th limbs in sketching posture and projected posture.  $\vec{ld_i^S}$  and  $\vec{ld_i^P}$  is computed as Formula 3.  $n$  is the number of limbs.

For a sequence of input sketching postures, what the user desires is a consecutive motion, in which the similar postures should appear in the same order as input postures. When given  $n$  sketching postures  $\{S_1, S_2, \dots, S_n\}$ , our posture by posture retrieval algorithm is presented in Algorithm 1.

In Algorithm 1, we utilize a pre-computed directed motion graph [7], which is similar as Peng's work [28], to splice some motion segments into a consecutive motion.

A clustering method for key postures extraction is proposed in Section 5.1. Here, we modify it to cluster all the postures in database. Namely, for the incoming posture, we compute the distance between the new posture and center postures of all existing cluster and decide its cluster. Each motion graph node represent a cluster in motion database. If two clusters are adjacent in a motion file, an edge between the corresponding graph nodes is added. As illustrated in Figure 6, three motions are clustered into five classes, then a directed motion graph with five nodes is created. A transition path 'C-A-D' is found when user requires a motion from node C to node D.

In the Step 6 of Algorithm 1, assume that  $k$  groups is included in  $U$ . After selecting one motion segment in each group,  $k$  motion segments  $MS_1, MS_2, \dots, MS_K$  are acquired. We have to find a transition path, which passes from  $MS_1$  to  $MS_K$  orderly, by traveling the motion graph. For arbitrary two neighbouring result segment  $MS_i$  and  $MS_{i+1}$ . Firstly we locate the two nodes that the last posture of  $MS_i$  and the first posture of  $MS_{i+1}$  belong to. Then, we find the shortest path between the two nodes in the motion graph.



**Figure 6.** Motion graph construction based on clustering. (a) is the result after clustering, the capital letters represent cluster label. (b) is the motion graph.

#### Algorithm 1 Retrieval algorithm

**Require:**  $n$  sketches:  $\{S_1, S_2, \dots, S_n\}$

**Ensure:** Set of motion segment group:  $U$

- 1: **Initialize**  $i = 1$ .
- 2: Search the k-d tree and return  $m_c$  candidate postures for  $S_i$ . After selection by the user,  $m$  postures  $\{p_i^1, p_i^2, \dots, p_i^m\}$  are left.
- 3: **Foreach**  $p_i^j$  ( $1 \leq j \leq m$ ), search  $S_{i+1} \sim S_n$  orderly in the motion  $m_i^j$ , which  $p_i^j$  lies in. Assume that  $S_{i+1} \sim S_k$  ( $i \leq k \leq n, k = i$  means none of  $S_{i+1} \sim S_n$  is found in  $m_i^j$ ) is found in  $m_i^j$ . Find the  $m_i^g$  ( $1 \leq g \leq m$ ), which maximize  $k$  (may be more than one motion are found).
- 4: Add the motion segments between  $SP_i$  and  $SP_k$  in all the motions  $m_i^g$  to a group (several postures before  $SP_i$  and after  $SP_k$  are also selected). Add the group to  $U$ . Where  $SP_i$  is the posture in a motion corresponding to the sketch  $S_i$ .
- 5: **If** ( $k < n$ ), **Set**  $i = k + 1$ , **Go to** Step 2.
- 6: If  $U$  only contains one group, the motion segments in the group is returned as the retrieval results. Otherwise, we select any one motion segment from each groups and splice them into a consecutive motion by motion graph, which is introduced in Section 6. All the synthetic motions are returned as final results.

After that, the path from  $MS_i$  to  $MS_{i+1}$  is obtained, which could synthesize a continuous motion [28].

## 7. EXPERIMENTAL RESULT

Our experiments perform on a desktop computer with a 3.10 GHz Intel-Core-i5 cpu, 4 GB RAM and Windows7 operating system.

To demonstrate the efficiency of our approach, we experiment on the CMU motion database [29], which contains 2434 motion data files. 222 584 projected-postures are extracted and a motion graph with 40 137 nodes is constructed.

A user study is conducted to show the effectiveness of our method. Ten participants, who are students and animators, are chosen in our experiments. They have a few minutes training before the user study. First, we will evaluate the accuracy of single posture retrieval. The participants sketching a posture and four similar candidate 3D posture results are given. If one of the results is accepted by the participant, we treat it as a successful retrieval. Each participant do it ten times with a different input posture each time. As shown in Table I, the accuracy of

**Table I.** Single posture retrieval result.

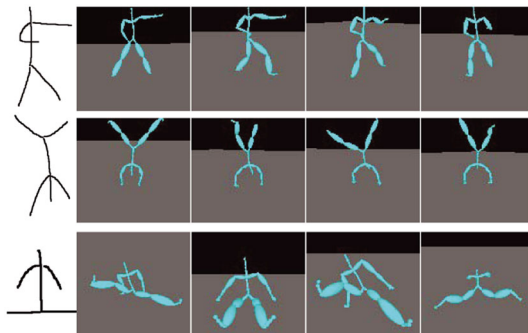
	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$	$P_{10}$
Success	9	10	10	8	10	9	10	8	9	10
Failure	1	0	0	2	0	1	0	2	1	0

$P_i$  is the  $i$ th participant.

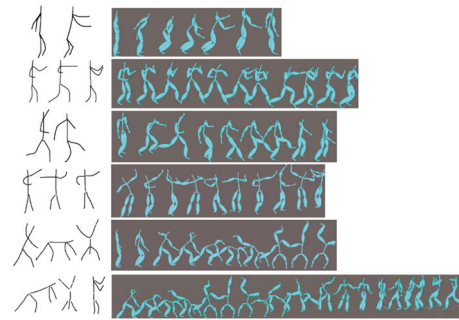


single posture retrieval is 93%. Some retrieval results are shown in Figure 7. The failure retrieval examples such as the third one in Figure 7 are mainly because the participant's sketching posture does not exist in the database. The average time to retrieve a single pose is about 1.5 s. If k-d tree index is not used, the average time is about 4.1 s for single pose retrieval.

Next, we evaluate the accuracy of motion sequence retrieval. Ten motions are picked randomly in the database. The participants retrieve them by sketching several key postures. If the candidate motion sequence results contain the target motion, we treat it as a successful retrieval. Each given motion will be displayed before the participants start to retrieve. The experiment earlier is repeated for ten times. Table II(a) shows the results. About three key postures inputs are needed if the user wants to retrieve a motion sequence. And about 87 s, which include the sketching time, are spent retrieving a motion. When the participant finishes drawing the key postures, our system can return the results in about 2.5 s. What is more, we ask each participant to retrieval arbitrary ten motions they want, which may not exist in the database. The result is evaluated by the participants like single posture retrieval earlier. It is shown in Table II(b). Figure 8 presents some examples of motion sequence retrieval, and the last one is synthesized by motion graph. Choi's [24] method is also evaluated in this phase. We gain the approximate



**Figure 7.** Examples of single posture retrieval in user study. The first two are successful and the third one is fail.



**Figure 8.** Examples of motion sequence retrieval in user study. The last result is synthesized by motion graph.

accuracy, but their retrieval average time is about 101 s, and the system response time for several sketching postures is about 8.2 s.

## 8. CONCLUSION

In this paper, we present a human motion retrieval framework that uses hand-drawn sketching key postures as query input. After the key posture extraction based on cluster and posture projection, a novel k-d tree index structure based on LD feature is constructed to index the motions in database, which is one of the main contributions in this work. Besides, an effective retrieval algorithm is proposed to get a consecutive motion result in the large motion database, which is another contribution. Our system can retrieve not only the existing motions in database but also the combined motion, which is spliced by some existing motions.

However, our retrieval framework still has some limitations. Firstly, a few complicated human motions such as martial arts motions and human interaction motions are difficult to be sketched by unprofessional user. Secondly, the posture by posture retrieval algorithm cannot handle the follow situation well. The user sketches three key postures and the third one has a big deviation. The algorithm may get two motion segments, where one segment contains the first two key postures and another segment contains the third key posture, and splice them into a motion. But

**Table II.** Motion sequence retrieval result.

	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$	$P_{10}$
(a)Retrieve the given motions										
Accuracy(%)	91	86	93	89	94	90	92	85	95	92
Sketch number <sup>1</sup>	2.6	2.9	3.1	2.3	3.9	2.7	2.8	3.3	3.5	3.1
Average time <sup>2</sup>	75	90	92	80	101	82	88	91	84	86
(b)Retrieve arbitrary motions										
Accuracy(%)	70	80	80	70	70	80	80	70	80	60

$P_i$  is the  $i$ th participant.

<sup>1</sup>The average number of input sketching postures.

<sup>2</sup>The average time, which includes sketching time, for retrieving one motion and the unit is seconds.

the third key posture that the user desired may be in the first segment. Thirdly, the trajectories of the joints, which may help retrieving more accurately, are not considered in this work.

## ACKNOWLEDGEMENTS

This research is supported by the National High Technology Research and Development Program (2012AA011502), the National Key Technology R&D Program (2013BAH59F00), the Zhejiang Provincial Natural Science Foundation of China (LY13F020001), the Zhejiang Province Public Technology Applied Research Projects (No. 2014C33090), the Fundamental Research Funds for the Central Universities(2014FZA5013), and partially supported by the grant of the Sino-UK Higher Education Research Partnership for PhD Studies Project funded by the Department of Business, Innovation and Skills of the British Government and Ministry of Education of China.

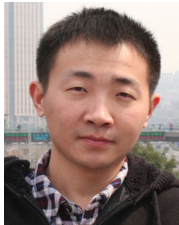
## REFERENCES

1. Böhm C, Berchtold S, Keim D. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys (CSUR)* 2001; **33**(3): 322–373.
2. Huang T, Liu H, Ding G. Motion retrieval based on kinetic features in large motion database. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ACM, Santa Monica, California, USA, 2012; 209–216.
3. Hecker C, Raabe B, Enslow R, DeWeese J, Maynard J, van Prooijen K. Real-time motion retargeting to highly varied user-created morphologies. In *ACM Transactions on Graphics (TOG)*, ACM, Los Angeles, USA, 2008; volume 27, 27.
4. Tak S, Ko H. A physically-based motion retargeting filter. *ACM Transactions on Graphics (TOG)* 2005; **24**(1): 98–117.
5. Sung M, Kovar L, Gleicher M. Fast and accurate goal-directed motion synthesis for crowds. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, Los Angeles, USA, 2005; 291–300.
6. Pullen K, Bregler C. Motion capture assisted animation: Texturing and synthesis. In *ACM Transactions on Graphics (TOG)*, ACM, San Antonio, USA, 2002; 501–508.
7. Kovar L, Gleicher M, Pighin F. Motion graphs. *ACM Transactions on Graphics* 2002; **21**(3): 473–482.
8. Jain S, Ye Y, Liu C-K. Optimization-based interactive motion synthesis. *ACM Transactions on Graphics (TOG)* 2009; **28**(1): 10.
9. Gleicher Michael. Motion editing with spacetime constraints. In *Proceedings of the 1997 Symposium on Interactive 3D graphics*, ACM, Providence, USA, 1997; 139–ff.
10. Min J, Liu H, Chai J. Synthesis and editing of personalized stylistic human motion. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, ACM, 2010; 39–46.
11. Yoshitaka A, Ichikawa T. A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering* 1999; **11**(1): 81–93.
12. Sun C, Junejo I, Foroosh H. Motion retrieval using low-rank subspace decomposition of motion volume. In *Computer Graphics Forum*, Wiley Online Library, Asheville, USA, 2011; volume 30, 1953–1962.
13. Qi T, Feng Y-F, Xiao J, Zhuang Y-T, Yang X-S, Zhang J-J. A semantic feature for human motion retrieval. *Computer Animation and Virtual Worlds* 2013; **24**(3–4): 399–407.
14. Müller M, Röder T, Clausen M. Efficient content-based retrieval of motion capture data. In *ACM Transactions on Graphics (TOG)*, ACM, Los Angeles, USA, 2005; volume 24, 677–685.
15. Numaguchi N, Nakazawa A, Shiratori T, Hodgins J-K. A puppet interface for retrieval of motion capture data. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, Wales, UK, 2011; 157–166.
16. Bentley J-L. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 1975; **18**(9): 509–517.
17. Chiu C-Y, Chao S-P, Wu M-Y, Yang S-N, Lin H-C. Content-based retrieval for human motion data. *Journal of Visual Communication and Image Representation* 2004; **15**(3): 446–466.
18. Deng Z-G, Gu Q, Li Q. Perceptually consistent example-based human motion retrieval. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*, ACM, Boston, USA, 2009; 191–198.
19. Zhu M-Y, Sun H-J, Deng Z-G. Quaternion space sparse decomposition for motion compression and retrieval. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 2012; 183–192.
20. Johnson M-P, Wilson A, Blumberg B, Kline C, Bobick A. Sympathetic interfaces: using a plush toy to direct synthetic characters. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, Pittsburgh, USA, 1999; 152–158.



21. Feng T-C, Gunawardane P, Davis J, Jiang B. Motion capture data retrieval using an artist doll. In *19th International Conference on Pattern recognition, 2008. ICPR 2008*, IEEE, 2008; 1–4.
22. Thorne M, Burke D, van de Panne M. Motion doodles: an interface for sketching character motion. In *ACM Transactions on Graphics (TOG)*, ACM, 2004; volume 23, 424–431.
23. Li Q-L, Geng W-D, Yu T, Shen X-J, Lau N, Yu G. Motionmaster: authoring and choreographing kung-fu motions by sketch drawings. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Vienna, Austria, 2006; 233–241.
24. Choi M-G, Yang K, Igarashi T, Mitani J, Lee J. Retrieval and visualization of human motion data via stick figures. In *Computer Graphics Forum*, Wiley Online Library, Hong Kong, China, 2012; volume 31, 2057–2065.
25. Lee J, Chai J-X, Reitsma P, Hodgins J-K, Pollard N-S. Interactive control of avatars animated with human motion data. In *ACM Transactions on Graphics (TOG)*, ACM, San Antonio, USA, 2002; volume 21, 491–500.
26. Min J-Y, Chen Y-L, Chai J-X. Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics (TOG)* 2009; **29**(1): 9.
27. Chao M-W, Lin C-H, Assa J, Lee T-Y. Human motion retrieval from hand-drawn sketch. *IEEE Transactions on Visualization and Computer Graphics* 2012; **18**(5): 729–740.
28. Peng J-Y, Lin I, Chao J-H, Chen Y-J, Juang G-H. Interactive and flexible motion transition. *Computer Animation and Virtual Worlds* 2007; **18**(4–5): 549–558.
29. Carnegie-mellon mocap database. (Available from: <http://mocap.cs.cmu.edu/>) [October 2012].

## AUTHORS' BIOGRAPHIES



**Zhangpeng Tang** got his bachelor's degree (2012) from the Department of Cognitive Science at Xiamen University (China). Now, he is a second year master's student majoring in computer science and technology at Zhejiang University (ZJU, China). His research fields include computer animation and machine learning.



and multimedia retrieval.

**Jun Xiao** got his PhD degree from the College of Computer Science at ZJU (2007) majoring in computer science and technology. Now, he is an associate professor of the College of Computer Science at ZJU, working in Microsoft Visual Perception Lab of ZJU. His research fields include computer animation, digital entertainment,



vision, and machine learning.

**Yinfu Feng** received his BS degree in Information Security from University of electronic science and technology of China, Chengdu, China, in July 2009. Now, he is a fourth year PhD student majoring in computer science and technology at ZJU. His current research interests include multimedia analysis and retrieval, computer



**Xiaosong Yang** is a senior lecturer in the National Centre for Computer Animation, The Media School, Bournemouth University, UK. He received his bachelor (1993) and master's degree (1996) in Computer Science from ZJU (China), PhD (2000) in computing mechanics from Dalian University of Technology (China). He worked as PostDoc (2000–2002) in the Department of Computer Science and Technology of Tsinghua University for 2 years, and research assistant (2001–2002) at the 'Virtual Reality, Visualization and Imaging Research Centre' of Chinese University of Hong Kong. His research interests include 3D modeling, animation, real-time rendering, virtual reality, virtual surgery simulation, and computer-aided design.



**Jian Zhang** is currently a professor of computer graphics at the National Centre for Computer Animation, Bournemouth University, UK, where leads the Computer Animation Research Centre. He is also a cofounder of the UK's Centre for Digital Entertainment, which received an initial funding of over six million GBP from the Engineering and Physical Sciences Research Council. His research focuses on a number of topics relating to 3D virtual human modelling, animation, and simulation, including geometric modelling, rigging and skinning, motion synthesis, deformation, and physics-based simulation.