

# A Semantic Feature for Human Motion Retrieval

## Abstract

With the explosive growth of motion capture data, it becomes very imperative in animation production to have an efficient search engine to retrieve motions from large motion repository. However because of the high dimension of data space and complexity of matching methods, most of existing approaches cannot return the result in real time. This paper proposes a high level semantic feature in a low dimensional space to represent the essential characteristic of different motion classes. Based on the statistic training of Gauss Mixture Model, this feature can effectively achieve motion matching on both global clip level and local frame level. Experiment results show our approach can retrieve similar motions with rankings from large motion database in real-time, and also can make motion annotation automatically on the fly.

**Keywords:** Motion retrieval, Motion annotation, Semantic feature, Key-frame extraction, Key-pose model

## 1 Introduction

With the development of motion capture techniques and the popularity of commercial systems such as multi-view cameras and Kinect, motion capture has become increasingly used in the production of featured films (such as Lord of the rings and Avatar), TV programmes and video games. In the past few years, lots of research has been focused on the manipulations of mocap data, such as motion synthesis [1–3], motion editing [4, 5] and motion retargeting [6] instead of the accumulated motion data itself. A large amount of human motion data has been

repetitively recorded and synthesised. However they are rarely reused for new movies and new games. Frequently, new motions will need to be captured for new animation designs. This waste of resources is because of the lacking of efficient search engine to retrieve similar motions from large motion repository.

In animation practice, the most commonly used motion retrieval approach use textual labels [7] such as "jumping" or "fighting" to search motion clips. Although text matching and retrieval can be processed very efficiently, textual descriptions cannot always sufficiently depict 3D motions and manual work is required in annotating motion segments in the repository. In the past few years, some research proposed to use sketched figures or trails to search the motion database. However because they [8, 9] require special training to the users and different artists may have different styles of drawing, it is very difficult to popularise this technique into the industry. So most research so far use a sample motion clip as an input query. There are two main challenges in these methods, how to construct a high-level feature that could describe a motion more efficiently and semantically, and how to calculate the similarity between two clips using a frame-based feature.

Intuitively, in a motion clip, we can always find some poses that can represent the whole clip, which we call key-poses. We assume that different groups of key-poses could describe different motion classes [10]. Based on these key-poses, in this paper, we propose a new semantic feature which is distinguishable enough for human motion retrieval. The process of feature calculation is shown as Figure 1. Firstly, we use k-means clustering algorithm to get the key-

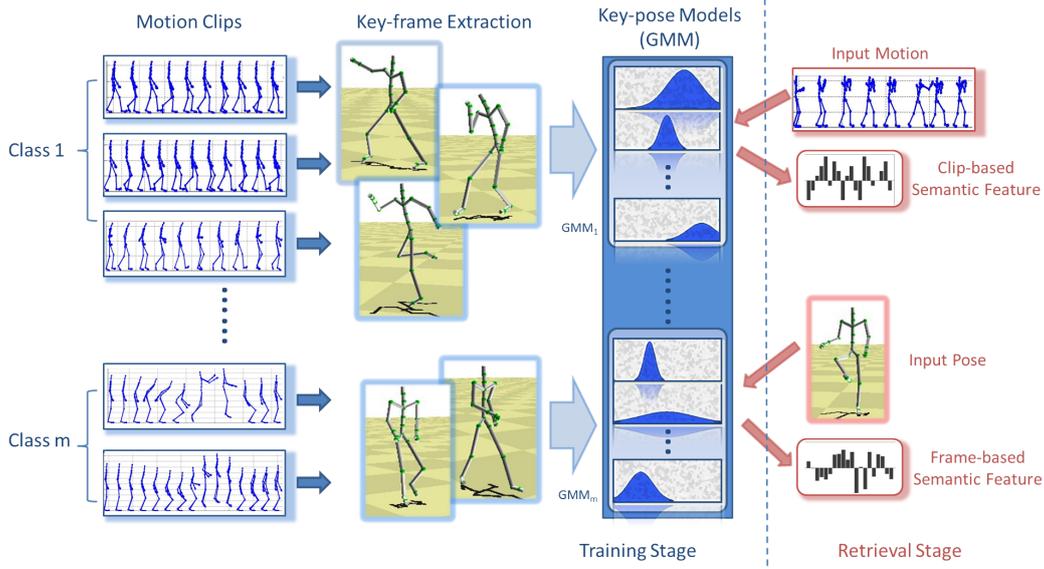


Figure 1: Flowchart: the construction (left side) and application (right side) of our semantic feature.

frame of each clip in the database. Secondly the key-pose model is constructed from all the key-frames of every motion class by Gauss Mixture Model (GMM). Then the frame-based and clip-based semantic feature can be extracted for all the motions in the motion database. In the retrieval procedure, given an input clip, Sparse Coding (SC) method is used to get the matched result from database based on the matching of semantic features. The other application of our feature is motion annotation. The frame-based semantic feature is calculated frame by frame, and then labelled with the max probability of each class. Experiment results show our approach can retrieve the similar motions with rankings from large motion database in real-time, and also can make motion annotation automatically on the fly.

To summarize, we have the following three main contributions:

1. A semantic feature based on Gaussian Mixture Model (GMM) is proposed. Based on the statistic training on all motion classes, this new feature represents more accurately the key characteristic of different motions.
2. The semantic feature is a frame-based feature, but can be transferred into a clip-based one, so that the similarity between two clips can be compared directly.
3. SC provides better matching result than traditional K Nearest Neighbor (KNN) method of

Euclidian distance, because the reconstruction results by SC tend to get similar clips within the same motion class.

## 2 Related Works

Liu et al. [11] has first proposed the concept of motion indexing and retrieval, which is now an important domain for the reuse of motion capture data in large-scaled database. Over the past few years, there exist many types of methods, which can be concluded as two categories, numeric-based retrieval and logic-based retrieval.

Most of the methods based on numerical similarity usually measure the distance between poses as a difference of the joint positions. As the development of motion capture devices, motion data becomes more precise, and the dimensionality gets higher. To avoid the curse of dimension, some dimension reduction methods has been introduced. Liu et al. [12] first apply PCA to the motion database, reduce the dimension of all the frames, and then generate a piecewise-linear model. Forbes et al. [13] employ weighted PCA to distinguish the different importance of different skeleton nodes. However, a serious problem is how to match two same motion clips but with different velocity. As a result, dynamic time warping (DTW) algorithm is applied for motion retrieval. Chiu et al.

[14] use pose-based index to identify the start and end frame of candidate motion clips, and then calculate the similarity by DTW. Similarly, many works of motion retrieval [13, 15] introduce DTW algorithm for comparing the similarity of two motions. However, DTW algorithm is very time-consuming, because too many distances must be calculated and compared.

Although the dimension reduction methods are widely used in motion retrieval, the compromise of losing details information and the accuracy of the matching is still an intractable problem. Therefore, some logic features are proposed to describe a motion, which can be summarized as two types, the frame-based feature [16–22] and the clip-based feature [8, 23–26].

Muller et al. [16] uses a set of Boolean geometric feature to express relations between several specific body joints. This method is extended later by Lin et al. [17], whose work is to build an index tree for motion retrieval, by selecting five featured bones and define three states for each. Another extension of geometric feature is the work of Chen et al. [18], in which an improved feature is presented to support partial similarity motion retrieval, and Adaboost is applied for feature selection. Other than geometric feature, Labanotation is also used as a frame-based feature. Yu et al. [19] generate a corresponding Labanotation sequence as additional motion property, then retrieve partly-matched motion segments and stitch them by a dynamic-programming-based method. Recently, Huang et al. [20] present kinetic interval features extracted from decomposed motion intervals, to describe the movement trend of motions. Because the motion retrieval is mainly clip-based, the local features are too dependent on DTW to solve the time-warping problem, while DTW is very time-consuming.

Another type of feature is clip-based feature. Sun et al. [24] apply multi-linear algebra tensor decomposition method to decompose the self-similar matrix of an action, and three feature vectors are extracted to express the spatial and temporal content of the movement. The trajectory of body joints can also be calculated as a feature, introduced by Pradhan et al. [25], and singular value decomposition is applied to the matrix, then the singular values and feature vectors is used as the descriptors for an

action. Kruger et al. [26] propose a kd-tree-based neighborhood search scheme, and use graph construction and path searching to match the global motions, which can be a substitute for subsequence DTW. In addition, frequently occurring temporal motion patterns (motion motifs) are introduced for retrieving motion data by Meng et al. [27]. In their work, the motif discovery problem is translated into finding continuous paths in a matching trellis, and a tree-growing method is introduced to search for the continuous paths. To conclude, the clip-based features mentioned above may not suffer from time-warping problem, but so far, most of them are low-level features that are directly calculated from the original joint positions, so they cannot well describe the motion in semantic level.

Recently, some new techniques have been introduced for motion retrieval, such as sketch-based interface search. This technique is first applied in several fields, including 3D modeling [28–30] and retrieval [31, 32] and then extends to motion data retrieval. Chao et al. [8] use a hand-drawn sketch as a query, which is taken as a body joints trajectory. Then spherical harmonic (SH) is used to encode the trajectory and retrieve it from database. In the work of Choi et al. [9], sketched stick figures are used as an input query, and with converting the motions in database to a sequence of stick figures, the motions can be retrieved directly. However, the sketch may not be easily drawn by a freshman without professional training, and the styles of different artists are probably not unified.

In this paper, we present a novel semantic feature based on Gauss Mixture Model (GMM). Although it is a frame-based feature, we can easily transfer it to a clip-based one by averaging the probabilities of all frames in a motion.

## 3 Semantic Feature

### 3.1 Key-frame Extraction

For an original input human motion clip  $s = \{f_1, f_2, \dots, f_N\}$ , each frame  $f_i$  contains the  $x, y, z$  coordinates of all skeleton nodes. In pre-processing, each pose (or frame) should be registered to the origin point by shifting the root node and fixing the orientation to the same direction.

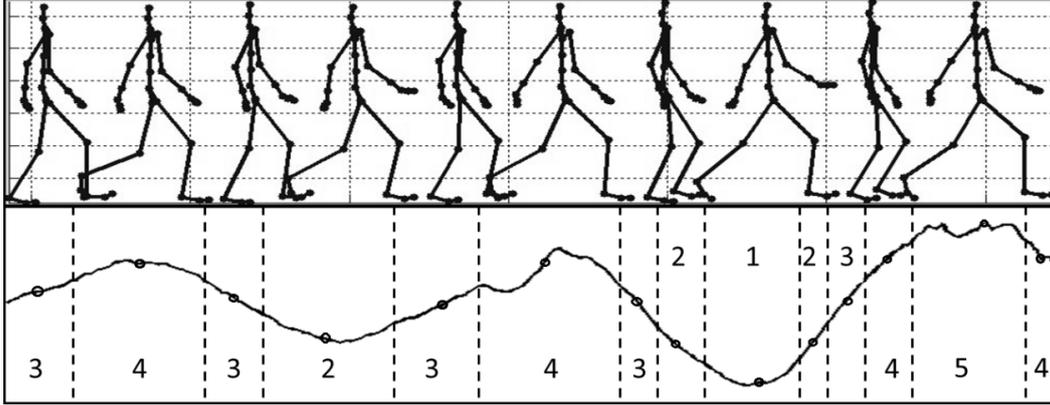


Figure 2: The illustration of key-frame extraction by k-means.

Some existing methods [33] perform well on key-frame extraction for a single motion, but the number of key-frames must be manually set in most of them. Considering the different velocity in motions, the number of key-frames should be adapted automatically, so they cannot work well for our problem. In this paper, a k-means based key-frame extraction method is proposed, which is inspired by the work of Liu et al. [11].

Firstly, k-means algorithm is applied to segment the input motion clip into many short subsequences. Then one frame of each subsequence is selected as the key frame. In detail, refer to Figure 2, assuming the curve in the bottom represents the motion clip in the top part, k-means algorithm is used to divide all frames into  $K$  clusters, and each frame is labeled by a cluster number (shown in Figure 2, corresponding to the motion curve). Then, the whole clip is segmented into short subsequences. Next, in each subsequence, the frame which is closest to their cluster centroid is selected as the key-frame of this subsequence (labeled as a circle in Figure 2). To avoid mislabel in very few frames, the subsequence that contains less than 5 frames will be ignored.

This method is automatic, because the number of key-frames will not be manually set for different motions since the number of clusters is determined. In addition, the result is stable regardless of the sampling rate or the velocity.

### 3.2 Key-pose Model Construction by GMM

Intuitively, we can find that the key-poses from a motion clip are more shared in the same motion class than in different classes. To differ from key-frames, key-poses are more commonly used for representing a specific kind of motion class, while a group of key-frames can only be used to represent a motion clip. Here, we try to construct the key-pose model for a specific kind of motion based on all the key-frames that extracted from the training motion clips that belong to the corresponding motion class.

In training time, all the motion clips are labeled by the tag of different pre-defined motion classes,  $\mathbf{C} = \{C_1, C_2, \dots, C_m\}$ . For a specific kind of motion class  $C_i$ , key frames set,  $\mathbf{X}_{C_i} = \{x_1, \dots, x_{N_{C_i}}\} (i = 1, \dots, m)$ , are extracted from the corresponding training motion clips, where  $N_{C_i}$  is number of key-frames extracted from the training motion clips that belong to the motion class  $C_i$ . We desire to build a suitable model based on the distribution probability of the extracted key-frames, which can calculate the probability of a given motion clip that belongs to each key-pose in  $C_i$ . As a result, in this paper, Gaussian Mixture Model (GMM) is applied to find the key-poses of each motion class based on the corresponding key-frame set  $\mathbf{X}_{C_i}$  and construct the probabilistic model.

We assume that there are  $P$  key-poses for each motion class,  $\mathcal{N}(\mu_k, \Sigma_k), k = 1, 2, \dots, P$ . EM algorithm is introduced to estimate the parameters in GMM iteratively. For each motion class  $C_i, i = 1, \dots, m$ , the initial priors for each

Gaussian model are set to equal,  $P(q_k|\Theta) = 1/P$ , where  $q_k$  represents each Gaussian model and  $\Theta$  is the parameter set. Then the probability of each data point  $x_n$  belongs to  $q_k$ ,  $P(q_k|x_n, \Theta)$ , can be calculated as:

$$\begin{aligned} P(q_k|x_n, \Theta) &= \frac{P(q_k|\Theta) \cdot p(x_n|q_k, \Theta)}{p(x_n|\Theta)} \\ &= \frac{P(q_k|\Theta) \cdot p(x_n|\mu_k, \Sigma_k)}{\sum_j P(q_j|\Theta) \cdot p(x_n|\mu_j, \Sigma_j)} \end{aligned} \quad (1)$$

To maximize the likelihood function  $\Theta^* = \text{argmax}_{\Theta} L(X|\Theta)$ , the parameters (means, variances and weights) are updated as:

$$\mu_k^{(new)} = \frac{\sum_{n=1}^{N_{C_i}} x_n P(q_k|x_n, \Theta)}{\sum_{n=1}^{N_{C_i}} P(q_k|x_n, \Theta)} \quad (2)$$

Let  $\bar{\mu}_k = (x_n - \mu_k^{(new)})$ ,

$$\Sigma_k^{(new)} = \frac{\sum_{n=1}^{N_{C_i}} P(q_k|x_n, \Theta) \cdot \bar{\mu}_k \cdot \bar{\mu}_k^T}{\sum_{n=1}^{N_{C_i}} P(q_k|x_n, \Theta)} \quad (3)$$

$$P(q_k^{(new)}|\Theta^{(new)}) = \frac{1}{N_{C_i}} \sum_{n=1}^{N_{C_i}} P(q_k|x_n, \Theta). \quad (4)$$

We repeat Equation 1 - 4 until the total likelihood increase for the training falls under some desired threshold. Then the key-pose model  $G_{C_i}$  for each motion class  $C_i$  is constructed, which can be used to estimate the probability of an input frame that belongs to the class  $C_i$ .

### 3.3 Semantic Feature Generation

Based on the above-mentioned key-pose model, two level semantic features can be extracted compactly.

The frame-based semantic feature  $t$  is composed of the probabilities  $p_{ik}$  that calculated by every single Gaussian model  $\mathcal{N}(\mu_k, \Sigma_k)$ ,  $k = 1, \dots, P$  of each key-pose model  $G_{C_i}$ ,  $i = 1, \dots, m$ :

$$t = [p_{11}, \dots, p_{1P}, \dots, p_{ik}, \dots, p_{m1}, \dots, p_{mP}], \quad (5)$$

where  $p_{ik}$  represents the probability of the current frame that belongs to the  $k$ th key-pose in class  $C_i$ .

This is a high level semantic feature that is represented by the key-poses of all the motion classes.

However, the annotation on poses is always inaccurate on some transitional poses between different actions or commonly used poses among different motion classes. In our method, the semantic feature used to describe the whole motion clip can also be easily generated based on our frame-based semantic feature.

For a given clip  $s = \{f_1, f_2, \dots, f_N\}$ , extract the key-frames set  $\tilde{s} = \{\hat{f}_1, \dots, \hat{f}_{n_s}\}$  by the method mentioned at 3.1, where  $n_s$  is the number of extracted key-frames  $\hat{f}_i$  in motion clip  $s$ . Then, for every key-frame  $\hat{f}_i$ , compute the the frame-based semantic feature

$$t^{(j)} = [p_{11}^{(j)}, \dots, p_{1P}^{(j)}, \dots, p_{ik}^{(j)}, \dots, p_{m1}^{(j)}, \dots, p_{mP}^{(j)}], \quad j = 1 \dots n_s. \quad (6)$$

then the clip-based feature can be generated as the mean value of all the key-frame-based semantic feature:

$$\begin{aligned} \bar{t} &= \sum_i t^{(j)} / n \\ &= [\bar{p}_{11}, \dots, \bar{p}_{1P}, \dots, \bar{p}_{ik}, \dots, \bar{p}_{mP}]. \end{aligned} \quad (7)$$

The mean value  $\bar{p}_{ik}$  tells the statistical probability of the extracted key-frames to each key-pose in a motion, so it is a clip-based feature, which is close to the semantic meaning. Our feature is very convenient to classify or annotate a pose or the whole motion clip.

## 4 Application

### 4.1 Motion Classification and Annotation

In this paper, the semantic feature we proposed is calculated by all the probabilistic key-pose models of each motion class. The class could be determined by choosing the class tag with the highest probability.

For the frame-based motion classification or annotation application, the frame-based semantic feature  $t$  of each pose is calculated as Equation 5. Then the corresponding semantic tag  $C_i$  is determined by calculating:

$$\hat{i} = \text{argmax}_i \left( \sum_k p_{ik} \right). \quad (8)$$

For the clip-based motion classification or annotation application, the clip-based semantic feature  $\bar{t}$  of the input motion clip  $s$  is calculated as Equation 6 and 7. Then the semantic tag  $C_i$  can be determined by calculating:

$$\hat{i} = \operatorname{argmax}_i \left( \sum_k \bar{p}_{ik} \right). \quad (9)$$

Because the Gaussian models are constructed off-line, the classification and annotation process can be executed in real-time.

## 4.2 Motion Retrieval

In clip-based motion retrieval application, the clip-based semantic feature is used to calculate the similarity between the query example and the existing motions in database. Instead of calculating the Euclidian distance between the semantic features directly, Sparse Coding is applied here to select the similar motion clips. We assume that the input query example will be reconstructed well by the more related samples. Actually, the same idea has been widely used in the pattern recognition field.

For a database with  $N$  motion clips, given a motion clip  $s = \{f_1, f_2, \dots, f_N\}$  and the key-pose models  $G_{C_j}, j = 1, \dots, m$ , our motion retrieval algorithm is described in detail as follow:

1. Extract the key-frames set  $\tilde{s} = \{\hat{f}_1, \dots, \hat{f}_{n_s}\}$  for the given motion  $s$ .
2. Calculate the global semantic feature  $\bar{t}$  by every key-pose models  $G_{C_j}, j = 1, \dots, m$ .
3. Use SC to get the selection vector  $x$ , by solving

$$x = \operatorname{argmin} \|\mathbf{D}x - \bar{t}\|_2^2 + \lambda \|x\|_1, \quad (10)$$

where  $\mathbf{D}$  is a  $d \times N_t$  matrix which contains the clip-based semantic features of all the motions in database,  $d$  is the dimensionality of feature  $\bar{t}$  and  $N_t$  is the number of motion clips in database. The l1-ls solver is applied to solve this equation.

4. Find the max  $K$  coefficients in  $x$ , ranking them and the corresponding motions are the retrieval result.

## 5 Experimental Result

All of our experiments are based on CMU Human Motion Dataset, which is a most widely

Motion Class	Size of Class
Basketball	14
Basketball gestures	16
Boxing	9
Climb ladder	8
Dance	31
Stretching	10
Jumping	18
Running	44
Sit	14
Soccer	9
Walking	75
Wave	10

Table 1: The composition of the dataset, including 4 basic motions (walking, running, sit and jumping), 3 sports (soccer, basketball and boxing), 4 activities (stretching, dance, wave, climb ladder) and 1 gesture (basketball gestures).

used motion capture database. We choose 12 motion classes which include locomotion and physic activities (there are more than 12 motion classes in CMU dataset, but some of them contain too less motions to construct key-pose models). The composition of the dataset is shown in Table 1. All experiments were executed on a computer with an Intel Core i5 2400 3.1GHz and 4GB of RAM.

### 5.1 Key-frame Extraction

In this paper, key-frame extraction is applied in order to provide candidate poses for key-pose models. In our key-frame extraction method, the cluster number  $K$  is the key parameter that determines the density of key-frames and further affects the quality of key-pose models. Figure 3 shows the relationship among the parameter  $K$  ( $x$  axis), the overall number of key-frames (right  $y$  axis) and the correct rate of clip-based motion classification (left  $y$  axis).

As shown in Figure 3, the dashed line represent that the density of key-frames rises along with the increment of parameter  $K$ . The correct rate increases when  $K$  is less than 6, because the key-frames set is getting more complete. However, it is unstable when  $K$  is larger than 10

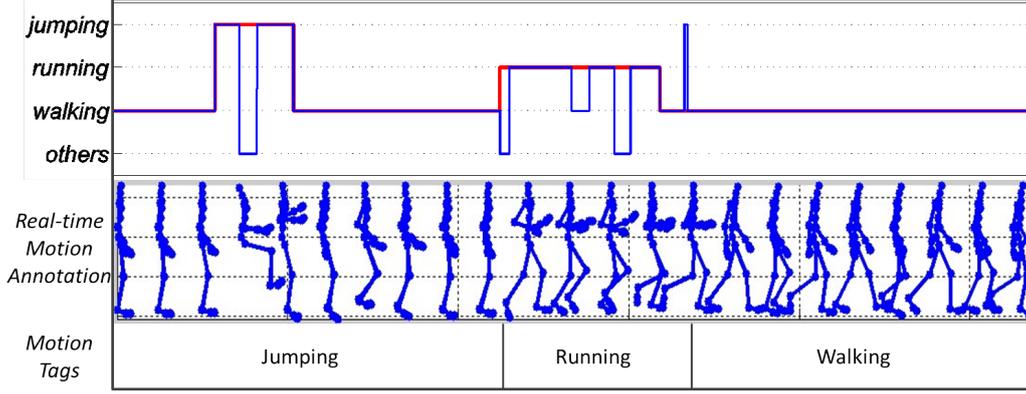


Figure 5: Motion annotation result

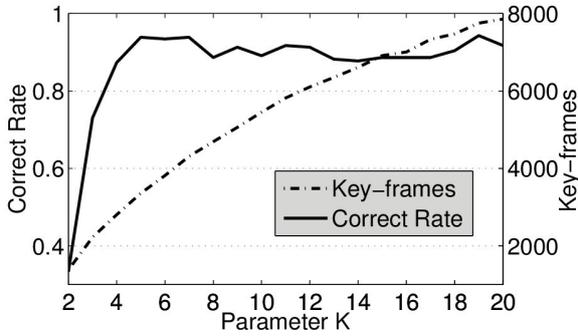


Figure 3: The influence of parameter  $K$ . The full line represents the classification correct rate with left  $y$  axis, while the dashed line is the number of extracted key-frames with right  $y$  axis.

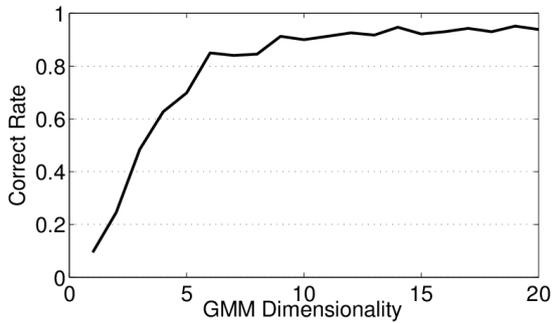


Figure 4: The classification correct rate with the GMM dimensionality

because the key-frames are too dense to distinguish with each other. As a result, the parameter  $K$  is set to be 7 in the following experiments.

## 5.2 Motion Classification

Clip-based motion classification is a key application to prove the effectiveness of our method. We randomly divide the dataset into two equal parts, one is served as training data and the other is served as testing data.

For constructing the key-pose models, the number of Gaussian models (parameter  $P$ ) in each GMM is another crucial parameter. Figure 4 shows the result that how parameter  $P$  affects the correct rate of motion classification. The performance is getting better along with the larger  $P$ , but the increment is unobvious when  $P$  is above 9, for the reason that the key-pose models are already complete enough. Considering the computing efficiency, the parameter  $P$  is set to be 9.

Real-time motion annotation is another application for our semantic feature. Several motion clips with different classes in the dataset are chosen and merged to be a long cross-category motion sequence. One of the annotation results is shown in Figure 5. The labeled class tags by our method are represented by blue line in top of the figure, while the original motion tags are in the bottom. It could be pointed out that the mislabeled frames are more probably appeared on some more commonly used poses between different motion classes. This is an unavoidable problem because such a pose cannot be distinguished without the context of the movement. A

simple solution for that is to smooth the curve and transfer the tags with short interval into the nearest one (represented by red line in top of Figure 5).

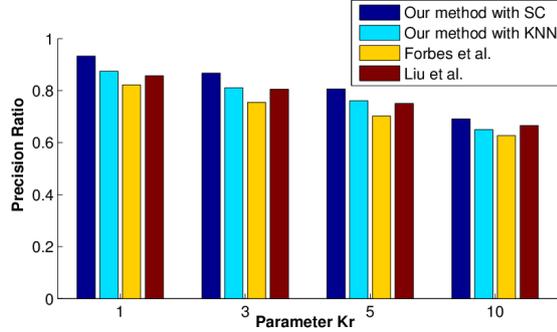


Figure 6: The comparison of retrieval precision among our method with SC and KNN, and other two methods.

### 5.3 Motion Retrieval

It is capable for our clip-based semantic feature to retrieve similar motion effectively and efficiently. The retrieval method SC and KNN and some other algorithms (Liu et al. [12] and Forbes et al. [13]) are compared in Figure 6, with the ranking list number  $K_r = 1, 3, 5$  and 10. The results show that the retrieval method SC is better than KNN in our method, and our method slightly outperforms the other two in precision ratio. We argue its main reason is that our key-pose model is more effective to describe the motions. In the method of Liu et al., a motion clip is indexed by the clustering result, while in our method, a clip is described by the probabilistic model, which contains more detailed information.

Table 2 shows the processing time of each step in both training and testing stage. We test the consumed time in three different size of dataset. The first dataset contains all the 258 motion clips with 425813 frames in 12 classes, the second contains 128 clips with 179166 frames in 10 classes and the last one contains 64 clips with 109066 frames in 8 classes. As shown in Table 2, the computing time is mainly spent in key-frame extraction, which is in training process. However, it is very fast in feature computation and motion retrieval, which are in

unit:sec

	Dataset1	Dataset2	Dataset3
Key-frame Extraction	787.01	314.5	178.25
Key-pose Construction	1.71	0.56	0.35
Frame-based Feature Computation	0.0010	0.0010	0.0009
Clip-based Feature Computation (Average)	1.63	1.48	1.32
Motion Retrieval (a single clip)	0.032	0.028	0.012

Table 2: Time consuming of the whole process

testing process. Moreover, it cost only 1.0 ms to compute the frame-based feature, so that it is fast enough for real-time motion annotation.

## 6 Conclusion

In this paper, a novel approach is proposed for human motion retrieval. After the key-frame extraction using k-means, the key-pose model is constructed based on GMM to calculate the probabilistic semantic feature, which is the main attribution in our method. The feature we proposed includes the frame-based feature that performs well on real-time motion annotation, and the clip-based feature that is appropriate for motion retrieval. At last, SC is introduced for the retrieval process instead of KNN, and 11-ls solver is applied to solve the sparse equation for the large-scaled database. The experiments show that our method is robust and efficient.

In the current approach, the key-pose models are stable unless a large number of changes to the database. If so, it will cause a recalculation on the key-pose models, but the key-frame extraction, which takes the most computation time, will be done only to the new motions. In addition, the optimal parameter  $P$  (discussed in 5.2) for each motion class will actually be different. For convenience, we fix the parameter  $P$  to a same large enough value, which satisfies most of the motion classes. It is proved by experi-

ment that the retrieval precision by a fixed large enough parameter  $P$  is almost the same as the precision by different optimal  $P$ s.

## References

- [1] Kovar L., Gleicher M., and Pighin F. Motion graphs. *ACM Transactions on Graphics (TOG)*, 21:473–482, 2002.
- [2] Jain S., Ye Y., and Liu C. Optimization-based interactive motion synthesis. *ACM Transactions on Graphics (TOG)*, 28:1–12, 2009.
- [3] Heck R. and Gleicher M. Parametric motion graphs. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 129–136, 2007.
- [4] Gleicher M. Motion editing with space-time constraints. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 139–ff, 1997.
- [5] Min J., Liu H., and Chai J. Synthesis and editing of personalized stylistic human motion. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, pages 39–46, 2010.
- [6] Hecker C., Raabe B., Enslow R., DeWeese J., Maynard J., and Prooijen K. Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics (TOG)*, 27, 2008.
- [7] Yoshitaka A. and Ichikawa T. A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11:81–93, 1999.
- [8] Chao M., Lin C., Assa J., and Lee T. Human motion retrieval from hand-drawn sketch. *Visualization and Computer Graphics, IEEE Transactions*, 18:729–740, 2012.
- [9] Choi M., Igarashi K., Mitani J., and Lee J. Retrieval and visualization of human motion data via stick figures. *Computer Graphics Forum*, 31:2057–2065, 2012.
- [10] Lv F. and Nevatia R. Single view human action recognition using key pose matching and viterbi path searching. In *Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [11] Liu F., Zhuang Y., Wu F., and Pan Y. 3d motion retrieval with motion index tree. *Computer Vision and Image Understanding*, 92:265–284, 2003.
- [12] Liu G., Zhang J., Wang W., and McMillan L. A system for analyzing and indexing human-motion databases. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 924–926, 2005.
- [13] Forbes K. and Fiume E. An efficient search algorithm for motion data using weighted pca. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 67–76, 2005.
- [14] Chiu C., Chao S., Wu M., Yang S., and Lin H. Content-based retrieval for human motion data. *Journal of Visual Communication and Image Representation*, 15:446–466, 2004.
- [15] Kovar L. and Gleicher M. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (TOG)*, 23:559–568, 2004.
- [16] Muller M., Roder T., and Clausen M. Efficient content-based retrieval of motion capture data. *ACM Transactions on Graphics (TOG)*, 24:677–685, 2005.
- [17] Lin Y. Efficient human motion retrieval in large databases. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 31–37, 2006.
- [18] Sun Z., Li Y., and Li Q. Partial similarity human motion retrieval based on relative geometry features. In *Digital Home (ICDH), 2012 Fourth International Conference*, pages 298–303, 2012.

- [19] Yu T., Shen X., Li Q., and Geng W. Motion retrieval based on movement notation language. *Computer Animation and Virtual Worlds*, 16:273–282, 2005.
- [20] Huang T., Liu H., and Ding G. Motion retrieval based on kinetic features in large motion database. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 209–216, 2012.
- [21] Naour T., Courty N., and Gibet S. Fast motion retrieval with the distance input space. In *Motion in Games Lecture Notes in Computer Science*, pages 362–365, 2012.
- [22] Muller M. and Roder T. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 137–146, 2006.
- [23] Deng Z., Gu Q., and Li Q. Perceptually consistent example-based human motion retrieval. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 191–198, 2009.
- [24] Sun C., Junejo I., and Horoosh H. Motion retrieval using low-rank subspace decomposition of motion volume. *Computer Graphics Forum*, 30:1953–1962, 2011.
- [25] Pradhan G. and Prabhakaran B. Indexing 3-d human motion repositories for content-based retrieval. *IEEE Transactions on Information Technology in Biomedicine*, 13:802–809, 2009.
- [26] Kruger B., Tautges J., Weber A., and Zinke A. Fast local and global similarity searches in large motion capture databases. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 1–10, 2010.
- [27] Meng J., Yuan J., Hans M., and Wu Y. Mining motifs from human motion. In *InEurographics 2008 C Short Papers*, pages 71–74, 2008.
- [28] Shin H. and Igarashi T. Magic canvas: interactive design of a 3-d scene prototype from freehand sketches. In *GI '07 Proceedings of Graphics Interface*, pages 63–70, 2007.
- [29] Olsen L., Samavati F., Sousa M., and Jorge J. Sketch-based modeling: A survey. *Computers & Graphics*, 33:85–103, 2009.
- [30] Gonen O. and Akleman E. Sketch based 3d modeling with curvature classification. *Computers & Graphics*, 36:521–525, 2012.
- [31] Eitz M., Richter R., Boubekeur T., and Hildebrand K. Sketch-based shape retrieval. *ACM Transactions on Graphics (TOG)*, 31, 2012.
- [32] Shao T., Xu W., Yin K., Wang J., Zhou K., and Guo B. Sketch-based shape retrieval. *Computer Graphics Forum*, 30:2011–2020, 2011.
- [33] Liu T. and Kender J. Computational approaches to temporal sampling of video sequences. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3, 2007.