Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Ensemble learning HMM for motion recognition and retrieval by Isomap dimension reduction[*]

XIANG Jian[†], WENG Jian-guang[†‡], ZHUANG Yue-ting, WU Fei

(*School of Computer Science*, *Zhejiang University*, *Hangzhou 310027*, *China*)

[†]E-mail: xiang_bj@cs.zju.edu.cn; wengjg@cs.zju.edu.cn

Received May 10, 2006; revision accepted Oct. 15, 2006

Abstract: Along with the development of motion capture technique, more and more 3D motion databases become available. In this paper, a novel approach is presented for motion recognition and retrieval based on ensemble HMM (hidden Markov model) learning. Due to the high dimensionality of motion's features, Isomap nonlinear dimension reduction is used for training data of ensemble HMM learning. For handling new motion data, Isomap is generalized based on the estimation of underlying eigen-functions. Then each action class is learned with one HMM. Since ensemble learning can effectively enhance supervised learning, ensembles of weak HMM learners are built. Experiment results showed that the approaches are effective for motion data recognition and retrieval.

Key words: Feature, Isomap, HMM (hidden Markov model), Ensemble learning, Motion recognition and retrieval
doi:10.1631/jzus.2006.A2063 **Document code:** A **CLC number:** TP37

INTRODUCTION

Now more and more motion capture systems are used to acquire realistic human motion data. Due to the success of the Mocap systems, realistic and highly detailed motion clips are commercially available and widely used for producing animations of human-like characters in a variety of applications, such as simulations, video games and animation files. Therefore an efficient motion data recognition and retrieval technique is needed to support motion data processing, such as motion morph, edition and synthesis, etc. At present, most motion data are stored in Mocap database with different length of motion clips, which is convenient for manipulating in animation authoring systems and retrieval based on keyword or content.

Conventional motion retrieval and processing systems based on motion capture data extract motion features from original data first, then some techniques like clustering algorithm are used to index motions with extracted features in database, and finally dynamic time warping algorithm (DTW) is used as a main method for measuring similarity between motions (Liu *et al*., 2003; Chiu *et al*., 2004; Muller *et al*., 2005; Zhai *et al*., 2003). For these systems, available index methods are not efficient and accurate enough to index motion data automatically and human motion type cannot be recognized by similarity measure processing.

In this paper, we describe a method which can recognize basic human actions like stand, walk and wave hands from motion clips automatically. The motion database can be indexed by HMM (hidden Markov model) learned automatically from training motion clips. Given a new motion, the system can recognize whether belongs to the trained motion classes. So this method can save computing time significantly in motion retrieval system and make an inverse retrieval system by using keyword (e.g. walk,

[‡] Corresponding author

[*] Project supported by the National Natural Science Foundation of China (Nos. 60533090 and 60525108), the National Basic Research Program (973) of China (No. 2002CB312101), and the Science and Technology Project of Zhejiang Province (Nos. 2005C13032 and 2005C11001-05), China

jump) instead of query example to implement motion retrieval. And understanding some motions' semantic automatically also is helpful in subsequent processing, such as motion retargeting, motion synthesis, etc.

In order to retrieve motion data accurately, the features used to represent motion data play a key role. Until now, several motion features have been proposed. Chiu *et al.*(2004) proposed local spherical coordinates relative to the root orientation as the segments posture of each skeletal segment; Liu *et al.*(2003) constructed a motion index tree based on a hierarchical motion description for motion retrieval; Mueller *et al.*(2005) introduced 31 Boolean features expressing geometric relations between certain body points of a pose. Here we use some motion aspects as features, such as joints positions, angles, speed, etc. Since 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 is called "Curse of dimensionality (Beyer *et al.*, 1999)", and high dimensional data also result in more computation overload at the same time. Therefore some dimensionality reduction techniques are necessary for motion retrieval. Leung and Li (2002) presented the architecture and facilities for providing Media-on-demand for an Agent-based Tutoring system (MATS).

The classical techniques of dimensionality reduction such as PCA, ICA, LDA and MDS are inherently linear methods of dimensionality reduction. LLE and Isomap are two representative nonlinear dimensionality reduction techniques. LLE (Roweis and Saul, 2000) finds *K* nearest neighbors of each point and computes the weights that best reconstruct each datapoint from its neighbors. So the intrinsic structure is preserved. Isomap (Tenenbaum *et al.*, 2000) constructs neighborhood graph and computes shortest path distances (geodesic distances) for each pair of points in the graph. Then classical MDS is used with geodesic distances. Because human motion data are very complex and have non-linear intrinsic structures that are invisible to PCA or MDS, here we use Isomap to map original motion clips into low-dimensional manifold.

As explained above, Isomap can embed a given set of samples to a low dimensional space to "maximally" preserve all pairs geo-distance. However, geo-distance of Isomap is only defined on training sets and raw Isomap cannot map new samples to the embedding space because it requires a whole set of points in the database to calculate geo-distance. So new queries outside the given database cannot be handled by raw Isomap. An incremental Isomap in (Law *et al.*, 2004) is proposed to treat increasing data, but the low-dimensional embedding of the training set must be recomputed every time. Shi *et al.*(2005) proposed an extension of Isomap to apply a trained model to new datapoints. But the extension still needs to compute geo-distances between new samples and training sets. An RBF (Radial Basis Function) neural network (He *et al.*, 2004) is trained to approximate the optimal mapping function from input space to the embedding space. In this paper, we use a generalization of Isomap (Bengio *et al.*, 2003). This generalization with a unified framework in which these algorithms are seen as learning eigenfunction of a kernel can yield embeddings for new samples.

When motion clip data is embedded to a low dimensional space, assume that this space's dimensionality is *d*. We can say that these *d* features are intrinsic features of motion. For each feature, the dynamics of one action class is learned from a continuous HMM with outputs modelled by a mixture of Gaussian distribution. HMM is a kind of temporal training model used successfully in speech recognition (Rabiner, 1989), and has been applied to video content analysis in constrained conditions (Starner, 1995; Liu and Chen, 2004). Yin *et al.*(2004) proposed an integration called "boosted HMM" for lip reading. Their approach is different in that they use AdaBoost at first to select frame level features and then use HMM to exploit long term dynamics. Lv and Nevatia (2006) use HMM to recognize and segment motion data. But they propose 141 motion features to train HMM model, and computing time is too long to be applied in practical use.

Then weak classifiers for each feature are formed based on the corresponding HMM observation probabilities. Considering that ensemble learning that trains multiple learners to solve a problem can effectively improve the generalization in supervised learning, weak HMM classifiers of these features with strong discriminative power are then selected and combined by ensemble learning in our method. To our knowledge, there are very few methods that have been explored to integrate HMM with ensemble learning by motion capture data. Ensemble HMM learning has recently attracted the attention of many researchers. During the past years, diverse ensemble learning algorithms have been developed, such as Bagging (Breiman, 1996), AdaBoost (Freund and Schapire, 1995), etc. In this paper, AdaBoost is used for ensemble HMM learning.

Finally we test our automatic method on a large collection of motion capture clips with a large variety of actions and compare the performance with that of other methods. The recognition and retrieval results are very good and the method shows tolerance to considerable amount of noise.

MOTION REPRESENTATION AND ISOMAP DIMENSIONALITY REDUCTION

Motion representation and feature extraction

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

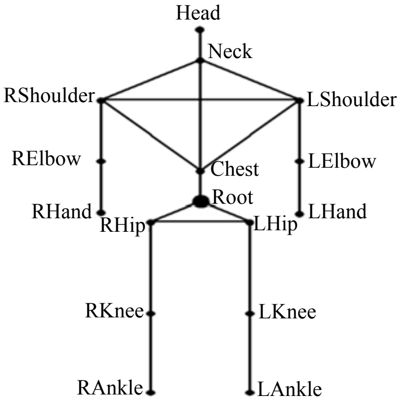

Fig.1 The skeleton model

World coordinate of each joint can be represented as follows:

$$M=\{F(1), F(2), \ldots, F(t), \ldots, F(n)\}, \tag{1}$$

$$F(t)=\{p(t), q_1(t), \ldots, q_m(t)\}, \tag{2}$$

where $F(t)$ is the tth frame in motion clip M, $p(t)$ is the rotation of the root joint, $q_i(t)$ is the rotation of joint i at frame t, m is the number of joints used in human skeleton.

All the motions used by us are performed by a real actor and recorded by an optical motion capture system at frame rate of 120. An actor wears a set of markers at each joint to identify the motion of the body joints. Optical motion capture systems (Mocap) triangulate the 3D position of a marker with a number of high precision cameras. The system produces motion data with 3 degrees of freedom for each marker, and rotational information must be inferred from the relative markers' orientation. We filter out the translation and rotation of root joint which present the overall position and orientation of the human skeleton and have no relation with specific movement. By motion capture system, each motion is presented by the same skeleton with 51 DOFs (corresponding to 16 joints of human body, see Fig.1). So in original motion data space, each frame of motion clip is represented as a vector with 48 dimensions.

Since each motion is a frame sequence, with each frame defining a posture, each posture is a configuration made up of all the body joints and each motion is a harmonic combination of sub-motion of all these joints, an efficient description of the posture is required. The description should also address the different effect of each joint on determining the posture. Hence we generate additional features of the skeletal motion by 16 joints of original motion data. In our implementation they include: (1) joint positions, (2) joint angles, (3) joint velocities. Similar motion features are used in (Assa *et al*., 2005). In our experiments, the above three aspects were found to be sufficient, whereas some other features such as joint angular velocities are found to be ineffective for motion recognition.

In our motion model, a human motion has 16 joints. We can calculate each joint's coordinate from the raw data. Based on this human skeleton, 8 bones in human limbs and a central bone that is connected by root and chest joints as a reference bone are extracted as the objects to represent motion feature. Each bone is defined as a vector from the upper joint to the lower joint in the human skeleton. Then bone angles are measured by using rotations of joints quaternion.

Joint velocity is approximated using the position differences between the pose before and after the given frame. In total, we have 72 features. A motion clip with n frames is represented as a vector with $n\times72$ dimensions. The data residing in such high-dimensional space results in much more computation time and has complex structure which is more difficult to analyze. So dimensionality reduction techniques should be introduced in the next section.

Isomap

The classical techniques of dimensionality reduction, PCA and MDS, are simple to implement, efficiently computable and guaranteed to discover the true structure of data lying on or near a linear subspace of the high-dimensional input space. PCA rotates the original feature space before projecting the feature vectors onto a limited number of axes. MDS aims to represent the datapoints in a lower dimensional space while preserving as much of the pairwise similarities between the datapoints as possible.

Here we use Isomap to do this job, which extends MDS by sophisticated distance measurement to achieve nonlinear embeddings. A graph of the data is built that is only locally connected within the neighborhoods of each point, and then the pairwise distances are measured by the length of the shortest path on that graph. This length is an approximation of the distance between its end points, as measured within the underlying manifold. Finally, MDS is used to find a set of low-dimensional points with similar pairwise distances. Isomap can not only reduce the dimensionality of high-dimensional input space, but also find meaningful low-dimensional structure hidden behind these original observations.

Three steps to implement Isomap are as follows:

(1) Construct neighborhood graph: define the graph $\boldsymbol{G}$ over all datapoints (in our method every point corresponds to a frame in the motion sequence) by connecting points i and j if (as measured by $d_x(i,j)$) they are closer than ε (ε-Isomap), or if i is one of the K nearest neighbors of j (K-Isomap). Set edge length to $d_x(i,j)$.

(2) Compute shortest paths: initialize $d_{\boldsymbol{G}}(i,j)=d_x(i,j)$ if i and j are linked by an edge; $d_{\boldsymbol{G}}(i,j)=\infty$ otherwise. Then for each value of $k=1, 2, \ldots, N$ in turn, replace all entry $d_{\boldsymbol{G}}(i,j)$ by $\min\{d_{\boldsymbol{G}}(i,j), d_{\boldsymbol{G}}(i,k)+d_{\boldsymbol{G}}(k,j)\}$. The matrix of final values $\boldsymbol{D_G}$ will contain the shortest path distances between all pairs of points in $\boldsymbol{G}$:

$$\boldsymbol{D_G}=\{d_{\boldsymbol{G}}(i,j)\}. \tag{3}$$

(3) Construct d-dimensional embedding: let λ_p be the pth eigenvalue (in decreasing order) of the matrix $\boldsymbol{T}(\boldsymbol{D_G})$, and v_p^i be the ith component of the pth eigenvector. Then set the pth component of the d-dimensional coordinate vector $\boldsymbol{y}_i$ equal to $\sqrt{\lambda_p}v_p^i$.

After non-linear reduction by Isomap (Fig.2), the low-dimensional embedding of the original motion clip is obtained with a very simple structure. Experiments showed that when 72 motion features are embedded to 7 or 8 low-dimensional space, residual error is minimal.

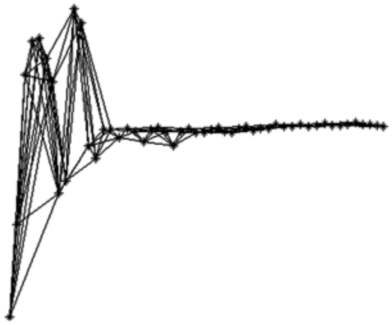

Fig.2 Low-dimensional manifold embedding of original motion clip

Generalization of Isomap

To obtain an embedding for a new datapoint, we need to generalize Isomap for new samples, which is learning the principal eigenfunction of a kernel (Bengio *et al.*, 2003) and the functions are from a function space whose scalar product is defined with the respect to a density model.

Let $D=\{x_1, x_2, \ldots, x_n\}$ be a dataset sampled from an unknown distribution with continuous density p and let P be the corresponding empirical distribution. Consider a Hilbert space $\hat{H}_p$ of functions with the following inner product:

$$\langle f,g\rangle_p=\int f(x)g(x)p(x)\mathrm{d}x,$$

where $p(x)$ is a weighting function.

So the kernel K can be associated with a linear operator K_p in $\hat{H}_p$:

$$(K_p f)(x)=\int K(x,y)f(y)p(y)\mathrm{d}y.$$

Now an "empirical" Hilbert space $\hat{H}_p$ is defined using the empirical distribution *P* instead of *p*. Let $\tilde{K}(a,b)$ be a kernel function that gives rise to a symmetric matrix $\tilde{\boldsymbol{M}}$ with entries $\tilde{M}_{ij}$=$\tilde{K}(x_i,x_j)$ upon *D*. Let ($\boldsymbol{v}_k$, λ_i) be an (eigenvector, eigenvalue) pair that solves $\tilde{K}_{\hat{p}} f_k = \lambda_k' f_k$ with *P* the empirical distribution over *D*. Let $e_k(x) = y_k(x)\sqrt{\lambda_k}$ or $y_k(x)$ denote the embedding associated with a new point *x*. Then:

$$\lambda_k' = \frac{1}{n}\lambda_k,\quad f_k(x) = \frac{\sqrt{n}}{\lambda_k}\sum_{i=1}^{n} v_{ik}\tilde{K}(x,x_i),$$

$$f_k(x_i) = \sqrt{n}v_{ik},\quad y_k(x)=\frac{f_k(x)}{\sqrt{n}}=\frac{1}{\lambda_k}\sum_{i=1}^{n} v_{ik}\tilde{K}(x,x_i),$$

$$y_k(x_i) = y_{ik},\ e_k(x) = \sqrt{\lambda_k}\,y_k(x),\ e_k(x_i) = e_{ik}.$$

The detail and further justifications of the above formulae can be seen in (Bengio *et al*., 2003).

Here the definition of $\boldsymbol{D}_{\boldsymbol{G}}(a,b)$ in Eq.(3) is used, which only uses the training points in the intermediate points on the path from *a* to *b*. We obtain a normalized kernel as follows:

$$\tilde{K}(a,b) = -\frac{1}{2}\{\boldsymbol{D}_{\boldsymbol{G}}(a,b) - E_x[\boldsymbol{D}_{\boldsymbol{G}}{}^2(x,b)] \\ - E_{x'}[\boldsymbol{D}_{\boldsymbol{G}}{}^2(a,x')] + E_{x,x'}[\boldsymbol{D}_{\boldsymbol{G}}{}^2(x,x')]\}.$$

Then the following formula is applied for the extension of Isomap to a new point *x*. It can yield the projection of *x* on the principal components of the corresponding low-dimensional datapoints.

$$e_k(x) = \frac{1}{2\sqrt{\lambda_k}}\sum_i v_{ik}\{E_{x'}[\tilde{D}^2(x',x_i)] - \tilde{D}^2(x_i,x)\}. \quad (4)$$

If we have access to a huge amount of data to estimate the eigenfuntions of K_P in $\hat{H}_p$, the optimal out-of-sample embedding would be obtained as a new sample. The consequence of the above results is that Isomap which only provided an embedding for the training examples can be extended to provide an embedding for new samples. So our system can handle new queries outside the given motion database by the generalization of Isomap.

ENSEMBLE HMM LEARNING

Weak HMM classifier

We choose an HMM to capture the dynamic information in the feature vectors as experience showed HMM to be more powerful than models such as Bayesian network or DTW. The basic theory of HMM was presented in the late 1960s and early 1970s. Widespread understanding and application of the theory of HMMs to speech processing occurred within the past several years.

An HMM is a quintuple (N, M, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{\pi}$), where N is the number of states in the model; M is the number of distinct observation symbols per state; the state transition probability distribution $\boldsymbol{A}$={a_{ij}}, a_{ij}=$P[q_{t_{n+1}}$=$j \mid q_{t_n}$=$i]$, 1≤i,j≤N, a_{ij}>0; the observation symbol probability distribution in state *j*, $\boldsymbol{B}$=$b_j(k)$, where $b_j(k)$=$P[o_t$=$v_k|q_t$=$j]$ (1≤k≤m); the initial state distribution $\boldsymbol{\pi}$={π_i}, where π_i=$P[q_1$=$i]$ (1≤i≤M).

There are three basic problems of interest that must be solved for the model to be useful in real-world applications. These problems are:

(1) Given the observation sequence O=$O_1O_2\ldots O_T$ and a model parameter λ, how to efficiently compute $P(O|\lambda)$, the probability of the observation sequence. This problem can be solved by the Forward-Backward procedure.

(2) Given the observation sequence O=$O_1O_2\ldots O_T$ and a model parameter λ, how to choose a corresponding state sequence Q=$Q_1Q_2\ldots Q_T$ which is optimal in some meaningful sense. This problem can be solved by the Viterbi algorithm.

(3) How to adjust the model parameters λ given O such that $P(O|\lambda)$ is maximized. This problem can be solved by the Baum-Welch algorithm.

Assume that *N* types of common motions are selected and Isomap embeds the original data space into *D*-dimensional space (it means we extract *D* intrinsic features of motion clip). Then each HMM is learned for each type of motions with its *j*th feature and corresponding parameters λ_i (i=1, …, N). In our method, each HMM has 3 hidden states and each state is modelled by a 3-component mixture of Gaussian distribution. By Baum-Welch algorithm, we can reestimate the parameters of each HMM.

At time T, the probability of D dimensional observation vector being in state j,

$$b_j(o_t)=\sum_{m=1}^{M_j} c_{jm} N(o_t;\ \mu_{jm};\ \Sigma_{jm}),\ \ 1\le j\le N,$$

where M_j is the number of mixture components of Gaussian component in a state, c_{jm} is weight of mth mixture component (Gaussian), μ_{jm} is the value of expectation for Gaussian probability distribution $N(o_t$; μ_{jm}; Σ_{jm}), covariance matrix is Σ_{jm}, the Gaussian distribution is expressed as follows:

$$N(o_t;\mu_{jm};\Sigma_{jm})=\frac{1}{\sqrt{(2\pi)^n}\,|\Sigma_{jm}|}\mathrm{e}^{-\frac{1}{2}(o_t-\mu_{jm})'\Sigma_{jm}^{-1}(o_t-\mu_{jm})}.$$

Now c_{jm}, μ_{jm}, Σ_{jm} is learning as follows:

$$\overline{c}_{jm}=\sum_{t=1}^{T}\gamma_t(j,m)\Bigg/\sum_{t=1}^{T}\sum_{t=1}^{M_j}\gamma_t(j,m),$$

where the numerator is transition times from state j of mth probability model in time T, the denominator is transition times from state j of all probability models in time T.

$$\overline{\mu}_{jm}=\sum_{t=1}^{T}[\gamma_t(j,m)o_t]\Bigg/\sum_{t=1}^{T}\gamma_t(j,m),$$

$$\overline{\Sigma}_{jm}=\sum_{t=1}^{T}[\gamma_t(j,m)(o_t-\mu_{jm})(o_t-\mu_{jm})]\Bigg/\sum_{t=1}^{T}\gamma_t(j,m),$$

where $\gamma_t(j,m)$ is the probability of mth Gaussian model being in state j at time t.

Until now, an N×D matrix of HMMs is formed by each feature of each type of motions and $HMM_{i,j}$ is the HMM model of ith motion type and jth feature and its corresponding parameters are $\lambda_{i,j}$. All HMMs in column j correspond to the classifier for feature j.

Given one observation sequence O, we compute $P(O|\lambda)$ for each HMM using the Forward-Backward algorithm. Motion type classification based on feature j can be solved by finding action class i that has the maximum value of $P(O|\lambda)$. As shown in Eq.(5):

$$Action(O)=\underset{i=1,\ldots,N}{\arg\max}[P(O|\lambda)]. \tag{5}$$

The training and classification algorithm of HMM classifiers are listed as follows:

```
Procedure: HMM classification algorithm
Input: $M$ motion samples $[(x_1, y_1), \ldots, (x_M, y_M)]$, $x_k$ is a clip with motion types $y_k$, $y_k\in\{1, \ldots, N\}$, $k=1, \ldots, M$, $N$ is the number of types of motions.
    An observation clip $O=O_1O_2\ldots O_T$
Output: $Motiontype(O)$
(1) Classify these samples into $N$ classes, with each class containing the same type of motion.
(2) For $i$=1 to $M$
Train HMM for each feature of each motion type (using Baum-Welch algorithm)
(3) For $j$=1 to $N$
 Compute $P(O|\lambda_{j,i})$
(4) Return $Motiontype(O)=\underset{i=1,\ldots,N}{\arg\max}[P(O|\lambda)]$.
```

Since our HMM mode has 3 states with a 3-component mixture of Gaussian distribution and Baum-Welch algorithm usually converges in less than 10 iterations, the complexity of all HMM models training is $O(DM)$, M is the total length of training samples of all motion types. And the complexity of classification is $O(NT)$. So the complexity of the whole procedure is $O(DM+NT)$.

We found that these HMM classifiers have good performance: for example, out system can correctly recognize 65% of 10 common motion types. Nevertheless, final recognition need much better performance, and the method of ensemble learning with HMM will be discussed in the next section.

Ensemble learning

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new datapoints by taking a (weighted) vote of their predictions. The original ensemble method is Bayesian averaging, but more recent algorithms include error-correcting output coding, bagging, and boosting. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. One of the most active areas of research in supervised learning is to study methods for constructing good ensembles of classifiers. The main discovery is that ensembles are often much more accurate than the individual classifiers that make them up.

The most popular ensemble learning method is the boosting (Russell and Norvig, 2002) implemented on weighted training sets. In weighted training sets, each example has its weight $w_j \geq 0$. When the weight is higher, the example is more important during learning. The following is the algorithm.

```
Procedure: AdaBoost
Input: examples, set of $N$ labelled examples $[(x_1, y_1), \ldots, (x_N, y_N)]$; $L$, a learning algorithm; $M$ is the number of hypotheses in the ensemble
Output: a weighted-majority hypothesis
Local variables: $\boldsymbol{w}$, a vector of $N$ example weights, initially $1/N$; $\boldsymbol{h}$, a vector of $M$ hypotheses; $\boldsymbol{z}$, a vector of $M$ hypothesis weights
for $m$=1 to $M$ do
    $\boldsymbol{h}[m]=L(examples, \boldsymbol{w})$
    $error$=0
    for $j$=1 to $N$ do
        if $h[m](x_j)=y_j$ then $w[j]=w[j]*error/(1-error)$;
    for $j$=1 to $N$ do
        if $h[m](x_j)\neq y_j$ then $error=error+w[j]$
    $\boldsymbol{w}=Normalize(\boldsymbol{w})$
    $\boldsymbol{z}[m]=\log(1-error)/error$
return $\boldsymbol{h}$, $\boldsymbol{w}$, $\boldsymbol{z}$;
```

For all motion clips, initialize w_j=1. When the first hypothesis is given, the weights of motion clips in wrong classes increase and the weights in correct class reduce. Then a new weighted training set is created and new hypothesis is given based on this new training set repeatedly.

For the ensemble HMMs, a methodology for assessing prediction quality is used to estimate our method. First we collect a large set of examples and divide it into two disjoints sets: the training set and the test set, then use the proposed method with training set as examples to generate a hypothesis H and measure the percentage of example in the test set that are correctly classified by H. At last, steps above are repeated for different sizes of training sets and different randomly selected training sets of each size.

Comparing ensemble HMMs with the individual HMM learners, Fig.3 shows that the performance of the ensemble HMMs is higher. Results showed that after combining 5 features of HMM learners by the AdaBoost algorithm, the final learner achieves a recognition rate of 93.2% on the motion type Run, showing the effectiveness of the algorithm.

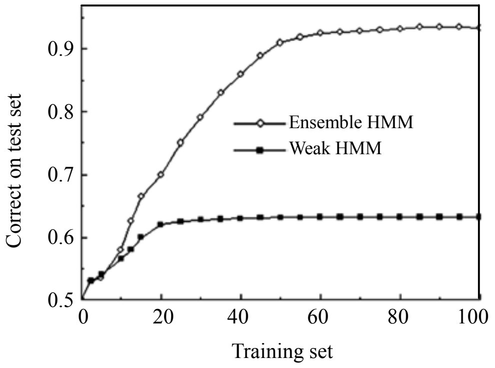

Fig.3 Comparison of the performance of ensemble HMM with weak HMM classifier

EXPERIMENTAL RESULTS AND ANALYSIS

We implement our algorithm in Matlab. It consists of more than 1000 motion clips with 15 common types in the database for test. Most of the typical human motions, such as walking, running, kicking, punching, jumping, washing floor, etc., are performed by actors.

Firstly, a kind of periodical motions Run, was experimented on. For example, in total we have 150 Run clips consisting of 38~79 frames, with the average number of frames being 56. So we can assume that, after features extraction, each frame can be represented by a 72-D vector and a 72×56 matrix. Then a 56×56 distance matrix is calculated by distances of each frame within the clip. So generalization of Isomap is used to reduce 72-D features to 7-D. Now 150 7×56 matrices are trained by ensemble HMM learning instead of 150 72×56 matrices and training time is saved significantly. Now a combination classifier of weak HMM classifiers for motion type Run is built. And so a classifier is trained for each common motion type.

We compare the performance of our proposed ensemble learning retrieval algorithm by Isomap dimension reduction with the same algorithm by linear approach PCA dimension reduction. Fig.4 shows the experimental result by looking at retrieval by learning different number of database. As can be seen, Isomap dimension reduction outperforms PCA with regard to motion data. For the motion, manifold is possibly highly nonlinear and PCA can only discover the linear structure.

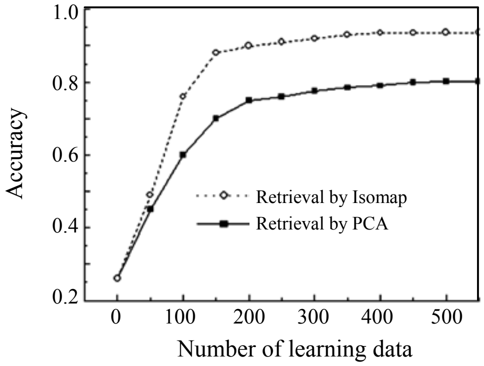

Fig.4 Comparison of retrieval by Isomap with retrieval by PCA

As discussed in Section 2.3, generalization of Isomap is learning the principal eigenfunction of a kernel to get a mapping formula from original data space to low-dimensional embeddings space. Because the optimal new samples embedding is only approximate, the retrieval performance by generalization of Isomap should not be better than that by raw Isomap. Fig.5 shows the retrieval performance by generalization of Isomap and raw Isomap. We use some new samples (100 motion clips) to do this comparing. For raw Isomap, geo-distances are recalculated using these new samples. As can be seen, the mapping formula of our generalization of Isomap can accurately approximate the manifold embedding of raw Isomap.

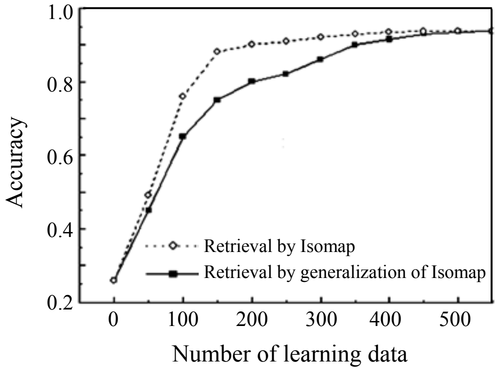

Fig.5 Comparison of retrieval by Isomap and generalization of Isomap

During motion recognition and retrieval, query examples always belong to common motion type. Given a query example, we compute $P(O|\lambda)$ for each motion type and found the type which has argmax($P(O|\lambda)$). So our motion retrieval avoids a great deal of motion similarity measuring and matching and is more efficient.

To compare the motion retrieval efficiency of the proposed method with that of the method based on Nearest Neighbor rule with dynamic clustering algorithm and DTW similarity measure (conventional method, which is described in (Liu *et al*., 2003)), Table 1 shows the retrieval time of these two methods. It is obvious that the time of our method is so much less than the time of the latter that the system performance has been improved significantly. The querying time by the conventional method depends greatly on the database size, but the time of our system is not related to it. Table 2 shows the comparison of retrieval efficiency between these two methods.

Table 1 Query time

Motion clips	$T1$		$T2$	
	N=200	N=800	N=200	N=800
A (43)	1.5412	4.5379	0.5161	0.7112
B (90)	1.8331	5.2412	0.8831	0.9413
C (150)	1.8912	5.6405	0.9953	1.0201
D (240)	2.2831	6.8019	1.2298	1.3217

$T1$ and $T2$ are query time of retrieval system with clustering index and with ensemble HMM learners, respectively

Table 2 Recall and precision

Motion clips	Recall		Precision	
	Conventional method	Our method	Conventional method	Out method
Walk	0.78	0.97	0.90	0.95
Run	0.70	0.94	0.83	0.96
Jump	0.51	0.91	0.70	0.91
Punch	0.43	0.88	0.56	0.90

Results showed that individual HMM learners have reasonably good performance; for example, individual HMM learner can correctly recognize 65.1% of motion Run and ensemble HMM learner can be clearly seen to have a gain of about 30% in the recognition rate (Table 3).

Table 3 Recognition rate of weak HMM and ensemble HMM

Motion	Recognition rate (%)	
	Weak HMM	Ensemble HMM
Walk	62.1	94.1
Run	64.9	95.2
Jump	66.5	93.3
Punch	64.9	91.0

Table 4 shows that Isomap dimension reduction before HMM learning can save a great deal of time for HMM training.

Table 4 Training time for HMM

Training data	Training time (s)	
	Original motion features	*D*-dimensional space by Isomap
Walk	56.3514	5.1266
Run	55.1191	6.2119
Jump	61.3133	5.5467
Punch	63.9681	5.2991

CONCLUSION AND FUTURE WORK

In this paper, an ensemble HMM learning method is proposed. Before learning, some motion features are extracted from motion data and generalization of Isomap is used to reduce dimension of these features and embed original motion space and new motion data into low-dimensional space. Then HMM models of some common motion types for each low-dimensional space feature are learned and ensemble learning method AdaBoost is applied to combine weak HMM learner for each feature to form strong learners for motion recognition. At last, the whole motion database is automatically built and efficiently and accurately indexed. The motion retrieval system is also sped up significantly.

In future, we will address the two following problems: (1) In order to get natural motion data efficiently, we will study motion sequence containing several different types of motions. How to retrieve short motion clip from long motion sequence with query motion requires further investigation. (2) In this paper, the research is based on motion capture data. If we can estimate joint positions like motion capture data from an image sequence or from video, our method can extend to a broad field, such as motion retrieval based on video, cross media, etc.

References

Assa, J., Caspi, Y., Cohen-Or, D., 2005. Action synopsis: Pose selection and illustration. *Proceedings of ACM SIGGRAPH*, **24**(3):667-676.

Bengio, Y., Paiement, J.F., Vincent, P., 2003. Out-of-Sample Extensions for LLE, ISOMAP, MDS, Eigenmaps, and Spectral Clustering. Proceedings of Neural Information Processing Systems, **16**:177-184.

Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U., 1999. When is “Nearest Neighbor” Meaningful? Proceedings of the International Conference on Database Theory, p.217-235.

Breiman, L., 1996. Bagging predictors. *Machine Learning*, **24**(2):123-140. [doi:10.1023/A:1018054314350]

Chiu, Y., Chao, S.P., Wu, M.Y., Yang, S.N., Lin, H.C., 2004. Content-based retrieval for human motion data. *Journal of Visual Communication and Image Representation*, **15**(3):446-466. [doi:10.1016/j.jvcir.2004.04.004]

Freund, Y., Schapire, R.E., 1995. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Proceedings of the 2th European Conference on Computational Learning Theory, p.23-37.

He, X.F., Ma, W.Y., Zhang, H.J., 2004. Learning an Image Manifold for Retrieval. Proceedings of 12th ACM International Conference on Multimedia, p.17-23. [doi:10.1145/1027527.1027532]

Law, M.H.C., Zhang, N., Jain, A.K., 2004. Nonlinear Manifold Learning for Data Stream. Proceedings of SIAM Data Mining, p.33-44.

Leung, E.W.C., Li, Q., 2002. Media-on-Demand for Agent-Based Collaborative Tutoring Systems on the Web. Proceedings of the IEEE Pacific Rim Conference on Multimedia, p.976-984.

Liu, F., Zhuang, Y.T., Wu, F., Pan, Y.H., 2003. 3D motion retrieval with motion index tree. *Computer Vision and Image Understanding*, **92**(2-3):265-284. [doi:10.1016/j.cviu.2003.06.001]

Liu, X.M., Chen, T., 2004. Video-Based Face Recognition Using Adaptive Hidden Markov Models. Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, p.340-345.

Lv, F.J., Nevatia, R., 2006. Recognition and Segmentation of 3-D Human Action Using HMM and Multi-class AdaBoost. 9th European Conference on Computer Vision (ECCV), **3954**:359-372.

Mueller, M., Roeder, T., Clausen, M., 2005. Efficient content-based retrieval of motion capture data. *Proceedings of ACM SIGGRAPH*, **24**(3):677-685.

Rabiner, L., 1989. A Tutorial on Hidden Markov Models and

Selected Applications in Speech Recognition. Proceedings of the IEEE, p.257-286.

Roweis, S.T., Saul, L.K., 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**(5500): 2323-2326. [doi:10.1126/science.290.5500.2323]

Russell, S., Norvig, P., 2002. Artificial Intelligence: A Modern Approach (Second Ed.). Prentice Hall, Englewood Cliffs, New Jersey.

Shi, L.K., He, P.L., Liu, B., 2005. A Robust Generalization of Isomap for New Data. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, p.1702-1712.

Starner, T., 1995. Visual Recognition of American Sign Language Using Hidden Markov Models. Master's Thesis. MIT Media Laboratory, p.189-194.

Tenenbaum, J., Silva, V.D., Langford, J., 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**(5500):2319-2323. [doi:10.1126/science.290.5500.2319]

Yin, P., Essa, I., Rehg, J.M., 2004. Asymmetrically Boosted HMM for Speech Reading. Proceedings of the IEEE Computer Society Conference, p.755-761.

Zhai, J., Yang, J., Li, Q., Liu, W.Y., Feng, B., 2003. Rich Media Retrieval on the Web—A Multi-level Indexing Approach. Proceedings of the 12th International World Wide Web Conference, p.383-390.