Learning Atomic Human Actions Using Variable-Length Markov Models

Yu-Ming Liang, Sheng-Wen Shih, Arthur Chun-Chieh Shih, Hong-Yuan Mark Liao, and Cheng-Chung Lin

***Abstract*—Visual analysis of human behavior has generated considerable interest in the field of computer vision because of its wide spectrum of potential applications. Human behavior can be segmented into atomic actions, each of which indicates a basic and complete movement. Learning and recognizing atomic human actions are essential to human behavior analysis. In this paper, we propose a framework for handling this task using variable-length Markov models (VLMMs). The framework is comprised of the following two modules: a posture labeling module and a VLMM atomic action learning and recognition module. First, a posture template selection algorithm, based on a modified shape context matching technique, is developed. The selected posture templates form a codebook that is used to convert input posture sequences into discrete symbol sequences for subsequent processing. Then, the VLMM technique is applied to learn the training symbol sequences of atomic actions. Finally, the constructed VLMMs are transformed into hidden Markov models (HMMs) for recognizing input atomic actions. This approach combines the advantages of the excellent learning function of a VLMM and the fault-tolerant recognition ability of an HMM. Experiments on realistic data demonstrate the efficacy of the proposed system.**

***Index Terms*—Atomic action learning, atomic action recognition, human behavior analysis, variable-length Markov models (VLMMs).**

I. Introduction

In RECENT years, visual analysis of human behavior has become a popular research topic in the field of computer vision. This is because it has a wide spectrum of potential applications, such as smart surveillance [7], [13], human computer interfaces [21], content-based retrieval [15], [22], and virtual reality [26]. Comprehensive surveys of related work can be found in [1], [10], and [24]. Wang *et al.* pointed out that a human behavior analysis system needs to address two low-level processes, namely, human detection and tracking and a high-level process of understanding human behavior [24]. While the low-level processes have been studied extensively, the high-level process has received relatively little attention. Human behavior usually consists of a series of atomic actions, each of which indicates a basic and complete movement. Since the human body is an articulated object with many degrees of freedom, inferring a body posture from a single 2-D image is usually an ill-posed problem. Providing a sequence of images might help to solve the ambiguity of behavior recognition. However, to integrate the information extracted from the images, it is essential to find a model that can effectively formulate the spatial–temporal characteristics of human actions. Note that if a continuous human posture can be quantized into a sequence of discrete postures, each one can be regarded as a letter of a specific language. Consequently, an atomic action composed of a short sequence of discrete postures can be regarded as a verb of that language. Sentences and paragraphs that describe human behavior can then be constructed, and the semantic description of a human action can be determined by a language modeling approach.

In a natural language, the most informative word of a sentence is usually its verb. Because an atomic action acts the verb of a sentence in a natural language, it is vital to recognize each atomic action in order to transform an input video sequence into semantic-level descriptions. Therefore, understanding human behavior involves the following two key issues: 1) how to segment the input video into clips of atomic actions and 2) how to recognize each segmented atomic action. Automatic segmentation of atomic actions is a popular research topic; a detailed survey of related research can be found in [6]. In this paper, we focus on the problem of automatic action recognition by using a language modeling approach to bridge the semantic gap between an atomic action sequence and a verb.

Language modeling [14], [20], a powerful tool for dealing with temporal ordering problems, has been applied in many fields, such as speech recognition [14], handwriting recognition [23], and information retrieval [8]. In this paper, we consider its application to the analysis of human behavior. A number of approaches have been proposed thus far. For example, Bobick and Ivanov [5] and Ogale *et al.* [16] used context-free grammars to model human actions, while Park *et al.* [17] employed hierarchical finite-state automata to recognize human behavior. In [27] and [28], hidden Markov models (HMMs) were applied to human action recognition. The HMM technique is useful for both human action recognition and human action sequence synthesis. Galata *et al.* [9] utilized variable-length Markov models (VLMMs) to characterize human actions, and showed that VLMMs trained with motion-capture data or silhouette images can be used to synthesize human action animations. Existing language modeling approaches for behavior analysis

Manuscript received September 10, 2007; revised March 20, 2008 and July 26, 2008. First published December 9, 2008; current version published January 15, 2009. This paper was recommended by Associate Editor F. Karray.

Y.-M. Liang and C.-C. Lin are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: ulin@iis.sinica.edu.tw).

S.-W. Shih is with the Department of Computer Science and Information Engineering, National Chi Nan University, Nantou 545, Taiwan (e-mail: swshih@ncnu.edu.tw).

A. C.-C. Shih is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: arthur@iis.sinica.edu.tw).

H.-Y. M. Liao is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan. He is jointly appointed by the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: liao@iis.sinica.edu.tw).

Digital Object Identifier 10.1109/TSMCB.2008.2005643

can be categorized into the following two classes: deterministic algorithms [5], [16] [17] and stochastic algorithms [9], [27], [28]. Since the latter have higher degrees of freedom than the former, they are suitable for a wider range of applications. Currently, the HMM is the most popular stochastic algorithm for language modeling because of its versatility and mathematical simplicity. However, since the states of an HMM are not observable, encoding high-order temporal dependences with this model is a challenging task. There is no systematic way to determine the topology of an HMM or even the number of its states. Moreover, the training process only guarantees a local optimal solution; thus, the training result is very sensitive to the initial values of the parameters. On the other hand, since the states of a VLMM are observable, its parameters can be estimated easily, given sufficient training data. Consequently, a VLMM can capture both long- and short-term dependences efficiently because the amount of memory required for prediction is optimized during the training process. However, thus far, the VLMM technique has not been applied to human behavior recognition directly because of the following two limitations: 1) It cannot handle the dynamic time warping problem, and 2) it lacks a model for handling the noise observation.

In this paper, we propose a hybrid framework of VLMM and HMM that retains the models' advantages while avoiding their drawbacks. The framework is composed of the following two modules: a posture labeling module and a VLMM atomic action learning and recognition module. First, a posture template selection algorithm is developed based on a modified shape context technique. The selected posture templates constitute a codebook, which is used to convert input posture sequences into discrete symbol sequences for subsequent processing. Then, the VLMM technique is applied to learn the symbol sequences that correspond to atomic actions. This avoids the problem of learning the parameters of an HMM. Finally, the learned VLMMs are transformed into HMMs for atomic action recognition. Thus, an input posture sequence can be classified with the fault tolerance property of an HMM.

The remainder of this paper is organized as follows. In Section II, we introduce the theory of VLMM. The proposed approach is described in Section III, and the experimental results are detailed in Section IV. Then, in Section V, we present our conclusions.

II. VLMM

A VLMM technique [9], [12], [19] is frequently applied to language modeling problems because of its powerful ability to encode temporal dependences. As shown in Fig. 1, a VLMM can be regarded as a probabilistic finite-state automaton (PFSA) $\Lambda = (S, V, \tau, \gamma, \pi)$ [19], where the variables are described as follows.

1) S denotes a finite set of model states, each of which is uniquely labeled by a symbol string representing the memory of a conditional transition of the VLMM,
2) V denotes a finite observation alphabet.
3) $\tau : S \times V \rightarrow S$ is a state transition function such that $\tau(s_j, \nu) \rightarrow s_{j+1}$.

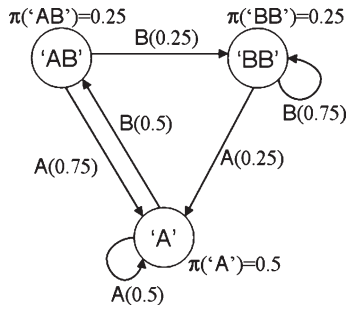


Fig. 1. Example of a VLMM.

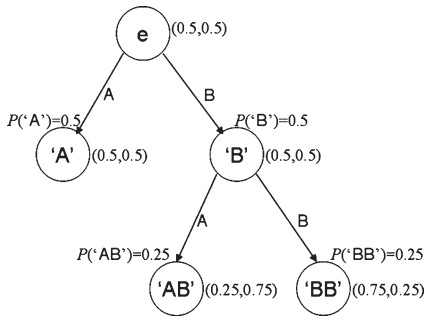


Fig. 2. PST for constructing the VLMM shown in Fig. 1.

4) $\gamma : S \times V \rightarrow [0, 1]$ represents the output probability function with $\forall s \in S, \sum_{\nu \in V} \gamma(s, \nu) = 1$.
5) $\pi : S \rightarrow [0, 1]$ is the probability function of the initial state satisfying $\sum_{s \in S} \pi(s) = 1$.

In the following sections, we consider the VLMM learning in Section II-A, followed by the VLMM recognition in Section II-B.

A. VLMM Learning

The topology and the parameters of a VLMM can be learned from training sequences by optimizing the amount of memory required to predict the next symbol. Usually, the first step of training a VLMM involves the construction of a prediction suffix tree (PST) [19]. A PST contains the information of the prefix of a symbol learned from the training data. Therefore, this prefix/suffix relationship helps one to determine the amount of memory required to predict the next symbol. After the PST is constructed from the training sequences, the PST is converted to a PFSA representing the trained VLMM. Fig. 2 shows the PST constructed from a training sequence for converting the VLMM shown in Fig. 1. Except for the root node, each node of the PST represents a nonempty symbol string, and each parent node represents the longest suffix tree of its child nodes. In addition, $P(\nu|s)$ is the output probability distribution of the next symbol ν of each node s that satisfies $\sum_{\nu \in V} P(\nu|s) = 1$. The output and prior probabilities can be derived from the training symbol sequences as follows:

$$P(\nu|s) = \frac{N(s\nu)}{N(s)} \tag{1}$$

$$P(s) = \frac{N(s)}{N_0} \tag{2}$$

where $N(s)$ is the number of occurrences of string s in the training symbol sequences and N_0 denotes the size of the training symbol sequences.

To optimize the amount of memory required to predict the next symbol, it is necessary to determine when the PST growing process should be terminated. Assume that s is a node with the output probability $P(\nu|s)$ and that $\nu's$ is its child node with the output probability $P(\nu|\nu's)$. We choose a termination criterion in order to avoid degrading the prediction performance of the reconstructed VLMM. Note that if the child node's output probability $P(\nu|\nu's)$ used to predict the next symbol ν is significantly better than the output probability $P(\nu|s)$ of the parent node, the child node is a deemed better predictor than the parent node; therefore, the PST should be grown to include the new child node. However, if the inclusion of a new child node does not improve the prediction performance significantly, the new child node should be discarded. Usually, the weighted Kullback–Liebler (KL) divergence is applied to measure the statistical difference between the probabilities $P(\nu|\nu's)$ and $P(\nu|s)$ as follows:

$$\Delta H(\nu's, s) = P(\nu's) \sum_{\nu} P(\nu|\nu's) \log \frac{P(\nu|\nu's)}{P(\nu|s)}. \quad (3)$$

If $\Delta H(\nu's, s)$ is greater than a given threshold, the node $\nu's$ is added to the tree. In addition to the KL divergence criterion, a maximal-depth constraint of the PST is imposed to further limit the PST's size. The PST contains all the information required to construct a PSFA, as shown in Figs. 1 and 2. The procedure for transforming a PST into its corresponding VLMM is described in [19].

B. VLMM Recognition

After a VLMM has been trained, it is used to predict the next input symbol according to a variable number of previously input symbols. In general, a VLMM decomposes the probability of a string of symbols $O = o_1 o_2 \dots o_T$ into the product of conditional probabilities as follows:

$$P(O|\Lambda) = \prod_{j=1}^{T} P(o_j | o_{j-d_j} \dots o_{j-1}, \Lambda) \quad (4)$$

where o_j is the jth symbol in the string and d_j is the amount of memory required to predict the symbol o_j.

The goal of VLMM recognition is to find the VLMM that best interprets the observed string of symbols $O = o_1 o_2 \dots o_T$ in terms of the highest probability. Therefore, the recognition result can be determined as model i^* as follows:

$$i^* = \arg\max_i P(O|\Lambda_i). \quad (5)$$

This method works well for natural language processing. However, since natural language processing and human behavior analysis are inherently different, two problems must be solved before the VLMM technique can be applied to atomic action recognition. First, as noted in Section I, the VLMM technique cannot handle the dynamic time warping problem; hence, VLMMs cannot recognize atomic actions when they are performed at different speeds. Second, the VLMM technique does not include a model for noise observation, so the system is less tolerant of image preprocessing errors. We describe our solutions to these two problems in the next section.

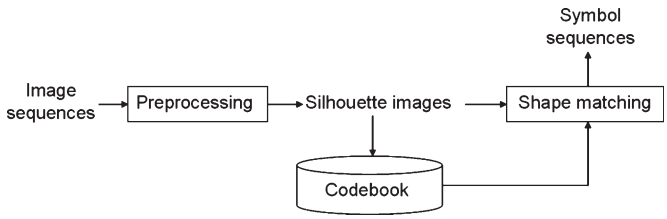


Fig. 3. Block diagram of the proposed posture labeling process.

III. Proposed Method for Atomic Action Recognition

The proposed method comprises the following two phases: 1) posture labeling, which converts a continuous human action into a discrete symbol sequence, and 2) application of the VLMM technique to learn the constructed symbol sequences and recognize the input posture sequences. The two phases are described in the following sections.

A. Posture Labeling

To convert a human action into a sequence of discrete symbols, a codebook of posture templates must be created as an alphabet to describe each posture. Although the codebook should be as complete as possible, it is important to minimize redundancy. Therefore, a posture is only included in the codebook if it cannot be approximated by existing codewords, each of which represents a human posture. In this work, a human posture is represented by a silhouette image, and a shape matching process is used to assess the difference between two shapes. Fig. 3 shows the block diagram of the proposed posture labeling process. First, a low-level image processing technique is applied to extract the silhouette of a human body from each input image. Then, the codebook of posture templates computed from the training images is used to convert the extracted silhouettes into symbol sequences. Shape matching and posture template selection are the most important procedures in the posture labeling process. These are discussed in the following sections.

1) Shape Matching With a Modified Shape Context Technique: We modified the shape context technique proposed by Belongie *et al.* [3] to deal with the shape matching problem. This modified method is aimed to improve the efficiency of posture labeling with the prerequisite of not sacrificing too much the labeling accuracy. In the original shape context approach, a shape is represented by a discrete set of sampled points $P = \{p_1, p_2 \dots p_n\}$. For each point $p_i \in P$, a coarse histogram h_i is computed to define the local shape context of p_i. To ensure that the local descriptor is sensitive to nearby points, the local histogram is computed in a log-polar space. An example of shape context computation and matching is shown in Fig. 4.

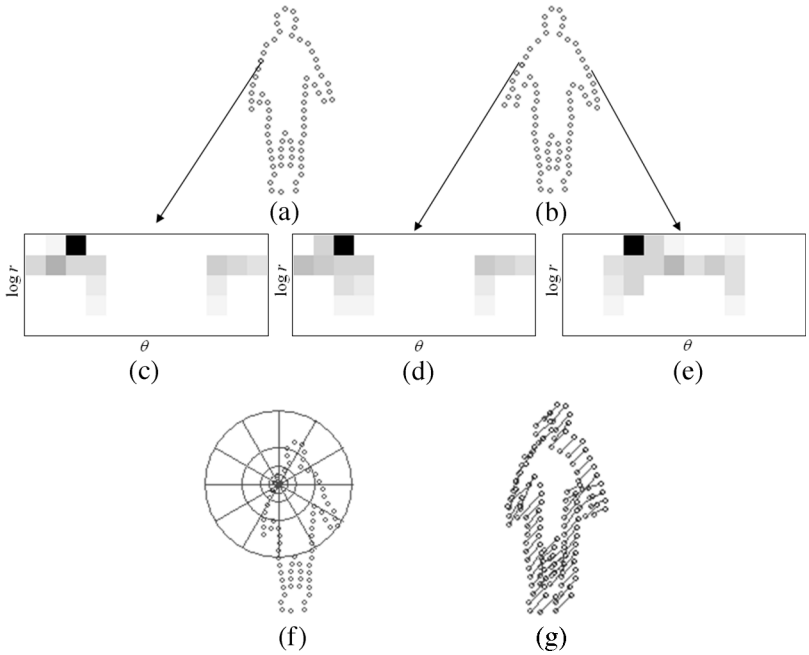


Fig. 4. Shape context computation and matching. (a) and (b) Sampled points of two shapes. (c)–(e) Local shape contexts corresponding to different reference points. (f) Diagram of the log-polar space. (g) Correspondence between points computed using a bipartite graph matching method.

Assume that p_i and q_j are points of the first and second shapes, respectively. The shape context approach defines the cost of matching the two points as follows:

$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \tag{6}$$

where $h_i(k)$ and $h_j(k)$ denote the K-bin normalized histograms of p_i and q_j, respectively. Shape matching is accomplished by minimizing the following total matching cost:

$$H(\pi) = \sum_i C\left(p_i, q_{\pi(i)}\right) \tag{7}$$

where π is a permutation of 1, 2, ..., n. Due to the constraint of one-to-one matching, shape matching can be considered as an assignment problem that can be solved by a bipartite graph matching method. A bipartite graph is a graph $G = (V = \{p_i\} \cup \{q_j\}, E)$, where $\{p_i\}$ and $\{q_j\}$ are two disjoint sets of vertices and E is a set of edges connecting vertices from $\{p_i\}$ to $\{q_j\}$. The matching of a bipartite graph is to assign the edge connection. There are many matching algorithms for bipartite graphs described in [2]. Here, the resulting correspondence points are denoted by $\{(p_i, q_{\pi(i)})|, i = 1, 2, \ldots, n\}$ or $\{(q_i, p_{\pi(i)})|, i = 1, 2, \ldots, m\}$, where n and m are the numbers of sample points on shapes P and Q, respectively. Therefore, the shape context distance between two shapes P and Q can be computed as follows:

$$D_{sc}(P, Q) = \frac{1}{n} \sum_i C\left(p_i, q_{\pi(i)}\right) + \frac{1}{m} \sum_j C\left(q_j, p_{\pi(j)}\right). \tag{8}$$

Although the shape context matching algorithm usually provides satisfactory results, the computational cost of applying it to a large database of posture templates is so high that is not feasible. To reduce the computation time, we only compute the local shape contexts at certain critical reference points, which should be easily and efficiently computable, robust against segmentation error, and critical to defining the shape of the silhouette. Note that the last requirement is very important because it helps preserve the informative local shape context. In this work, the critical reference points are selected as the vertices of the convex hull of a human silhouette. Matching based on this modified shape context technique can be accomplished by minimizing a modified version of (7) as follows:

$$H'(\pi) = \sum_{p \in A} C\left(p, q_{\pi(p)}\right) \tag{9}$$

where A is the set of convex hull vertices and H' is the adapted total matching cost. However, reducing the number of local shape contexts to be matched will also increase the influence of false matching results. To minimize the false matching rate, the ordering constraint of the vertices has to be imposed. However, since traditional bipartite graph matching algorithms [2] do not consider the order of all sample points, they are not suitable for our algorithm. Therefore, dynamic programming is adopted in the shape matching process. Suppose a shape P includes a set of convex hull vertices A and another shape Q includes a set of convex hull vertices B. The *convex-hull shape context* (CSC) distance can be calculated as follows:

$$D_{\text{csc}}(P, Q) = \frac{1}{|A|} \sum_{p \in A} C\left(p, q_{\pi(p)}\right) + \frac{1}{|B|} \sum_{q \in B} C\left(q, p_{\pi(q)}\right). \tag{10}$$

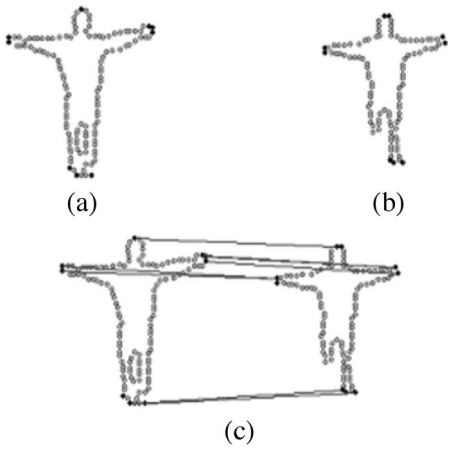


Fig. 5. Convex-hull shape context matching. (a) and (b) Convex hull vertices of two shapes. (c) Correspondence between the convex hull vertices determined using shape matching.

An example of CSC matching is shown in Fig. 5. There are three important reasons why CSCs can deal with the posture shape matching problem effectively. First, since the number of convex hull vertices is significantly smaller than the number of whole shape points, the computation cost can be reduced substantially. Second, convex hull vertices usually include the tips of human body parts; hence, they can preserve more salient information about the human shape, as shown in Fig. 5(a). Third, even if some body parts are missed by human detection methods, the remaining convex hull vertices can still be applied to shape matching due to the robustness of computing the convex hull vertices, as shown in Fig. 5.

2) Posture Template Selection: Posture template selection is used to construct a codebook of posture templates from training silhouette sequences. Here, we propose an automatic posture template selection algorithm (see Algorithm 1) based on the CSC discussed in Section III-A1. In the posture template selection method, the cost of matching two shapes [see (10) is denoted by $D_{\text{csc}}(b_i, a_j)$]. We only need to empirically determine one threshold parameter τ_C in our posture template selection method. This parameter determines whether a new training sample should be incorporated into the codebook. The selection of τ_C is not unique for all cases. Because incoming action sequences may contain any kind of action, the selection of τ_C is basically an ill-posed problem in mathematics. Therefore, we cannot determine a universal τ_C to fit in all cases. In fact, the selection of τ_C is not a major concern in this paper because our objective is to establish an automatic posture template selection scheme.

Algorithm 1: Posture Template Selection
Codebook of posture templates: $A = \{a_1, a_2, \ldots, a_M\}$
Training sequence: $T = \{t_1, t_2, \ldots, t_N\}$
for each $t \in T$ do {
  if ($A = \phi$ or $\min_{a \in A} D_{\text{csc}}(t, a) > \tau_C$) {
    $A \leftarrow A \cup \{t\}$
    $M \leftarrow M + 1$
  }
}

B. Human Action Sequence Learning and Recognition

Using the codebook of posture templates, an input sequence of postures $\{b_1, b_2, \ldots, b_n\}$ can be converted into a symbol sequence $\{a_{q(1)}, \ldots, a_{q(n)}\}$, where $q(i) = \arg\min_{j \in \{1,2,\ldots,M\}} D_{\text{csc}}(b_i, a_j)$. Thus, atomic action VLMMs can be trained by the method outlined in Section II-A. These VLMMs are actually different-order Markov chains. For simplicity, we transform all the high-order Markov chains into first-order Markov chains by augmenting the state space. For example, the probability of a d_ith-order Markov chain with state space S is given by

$$P(X_i = r_i | X_{i-d_i} = r_{i-d_i}, X_{i-d_i+1} = r_{i-d_i+1}, \ldots, X_{i-1} = r_{i-1}) \tag{11}$$

where X_i is a state in S. To transform the d_ith-order Markov chain into a first-order Markov chain, a new state space is constructed such that both $Y_{i-1} = (X_{i-d_i}, \ldots, X_{i-1})$ and $Y_i = (X_{i-d_{i+1}-1}, \ldots, X_i)$ are included in the new state space. As a result, the high-order Markov chain can be formulated as the following first-order Markov chain [11]:

$$\begin{aligned} &P(X_i = r_i | X_{i-d_i} = r_{i-d_i}, X_{i-d_i+1} \\ &\quad = r_{i-d_i+1}, \ldots, X_{i-1} = r_{i-1}) \\ &\quad = P\left(Y_i = (r_{i-d_{i+1}+1}, \ldots, r_i) | Y_{i-1} = (r_{i-d_i}, \ldots, r_{i-1})\right). \end{aligned} \tag{12}$$

Hereafter, we assume that every VLMM has been transformed into a first-order Markov model.

Next, we must solve the dynamic time warping problem and the lack of a model for noise observation problem. Note that the speed of the action affects the number of repeated symbols in the constructed symbol sequence: A slower action produces more repeat symbols. To eliminate this speed-dependent factor, the input symbol sequence is preprocessed to merge repeated symbols. VLMMs corresponding to different atomic actions are trained with preprocessed symbol sequences similar to the method proposed by Galata *et al.* [9]. However, this approach is only valid when the observed noise is negligible, which is an impractical assumption. The recognition rate of the constructed VLMMs is low because image preprocessing errors may identify repeated postures as different symbols. To incorporate a noise observation model, the VLMMs trained with unrepeated sequences must be modified to recognize input sequences with repeated symbols. Let a_{ij} denote the state transition probability from state i to state j. Initially, $a_{ii}^{\text{old}} = 0$ because the training data contain no repeated symbols. The self-transition probability is updated by $a_{ii}^{\text{new}} = P(\nu_i|\nu_i) + \delta$, where $P(\nu_i|\nu_i) = N(\nu_i\nu_i)/N(\nu_i)$ computed with the original training sequences and δ is a small positive number to prevent the overfitting problem [18]. Note that if the self-transition probability is zero, then an action sequence that contains repetition will result in a zero probability such that the system will not perform normally when faced with slower action sequences. To overcome this limitation, we add the small positive number δ to the self-transition probability. This parameter can be determined using the cross-validation method. The other

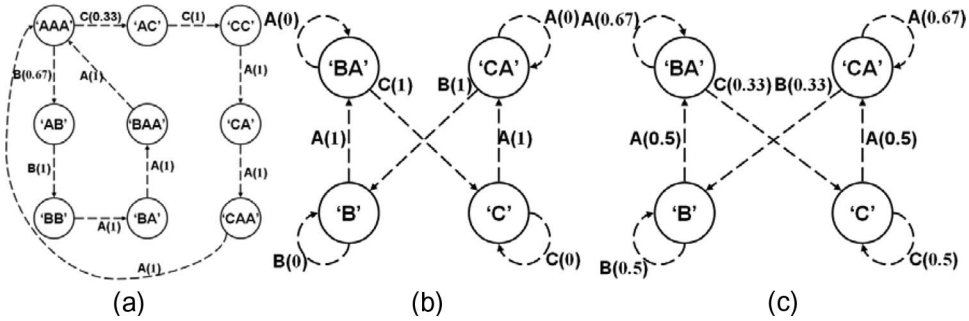


Fig. 6. (a) VLMM constructed with the original input training sequence. (b) Original VLMM constructed with the preprocessed training sequence. (c) Modified VLMM, which includes the possibility of self-transition.

transition probability must also be updated as $a_{ij}^{\text{new}} = a_{ij}^{\text{old}}(1 - a_{ii}^{\text{new}})$. For example, if the input training symbol sequence is "AAABBAAACCAAABB," the preprocessed training symbol sequence becomes "ABACAB." The VLMM constructed with the original input training sequence is shown in Fig. 6(a), while the original VLMM and the modified VLMM constructed with the preprocessed training sequence are shown in Fig. 6(b) and (c), respectively.

Next, a noise observation model is introduced to convert a VLMM into an HMM. Note that the output of a VLMM determines its state transition and vice versa because the state of a VLMM is observable. In general, the possible output is restricted to several discrete symbols. However, due to the noise caused by image preprocessing, the symbol sequence corresponding to an atomic action includes some randomness. Such randomness will cause the action sequence not recognizable by the VLMMs. Therefore, we propose to modify the symbol observation model as described in the following. Suppose that the output symbol of a VLMM is q_t at time t and that its posture template retrieved from the codebook is a_{q_t}. If the VLMM is the right model, the extracted silhouette image o_t will not deviate too much from its corresponding posture template a_{q_t}, provided that the segmentation result does not contain any major errors. Due to noise observation, the silhouette image o_t is a random variable, and so is the CSC distance $D_{\text{csc}}(o_t, a_{q_t})$. It is possible to learn the distribution of the CSC distance $D_{\text{csc}}(o_t, a_{q_t})$ using the training data. An example is shown in Fig. 7. In this example, it is clear that a Gaussian distribution can be applied to model the CSC distance, i.e., $P(o_t|q_t, \Lambda) = (1/\sqrt{2\pi}\sigma)e^{-D_{\text{csc}}(o_t,a_t)/2\sigma^2}$. The standard deviation σ of this distribution is estimated using the maximum-likelihood technique.

Note that the VLMM has now been converted into a first-order Markov chain. If the VLMM's observation model is detached from the symbol of a state, then the VLMM becomes a standard HMM. The probability of the observed silhouette image sequence $O = o_1 o_2 \ldots o_T$ for a given model Λ can be evaluated by the HMM forward/backward procedure with proper scaling [18]. Finally, category i^* obtained with the following equation is deemed to be the recognition result:

$$i^* = \arg\max_i \log\left[P(O|\Lambda_i)\right]. \tag{13}$$

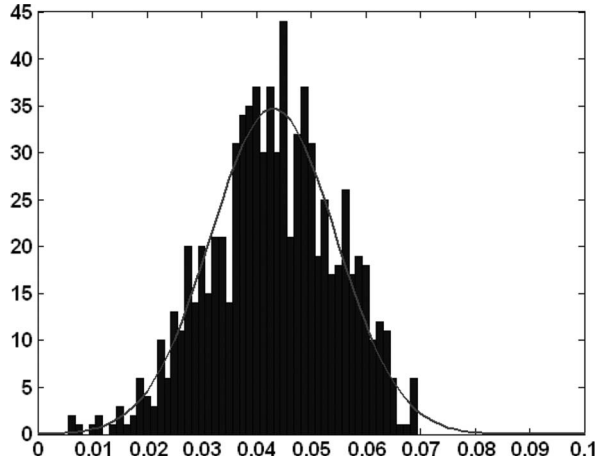


Fig. 7. Distribution of observation error, obtained using the training data.

IV. Experiments

We conducted a series of experiments to evaluate the effectiveness of the proposed method. A powerful scalable recognition system would only use the data extracted from one person for training but would still be capable of recognizing data collected from other people. Accordingly, the training data used in our experiments were a real video sequence comprised of approximately 900 frames [30]. The training data contained ten categories of action sequences that were performed by a single person. Some typical image frames are shown in Fig. 8. Using the posture template selection algorithm, a codebook of 95 posture templates (see Fig. 9) was constructed from the training data. The data were then used to build ten VLMMs, each of which was associated with one of the atomic actions shown in Fig. 8.

To demonstrate the effectiveness and efficiency of the proposed CSC matching process, we compared the posture labeling results and computation time of the proposed method with those of the original shape context matching approach. In the first experiment, the codebook of posture templates (see Fig. 9) was used to label the training data. The experimental results show that 85% of the labeling results done by the proposed method were the same as those obtained by applying the original method, but our approach could save 95% of the computation time. Since the computation time is very important in the procedure of human action recognition, a significant

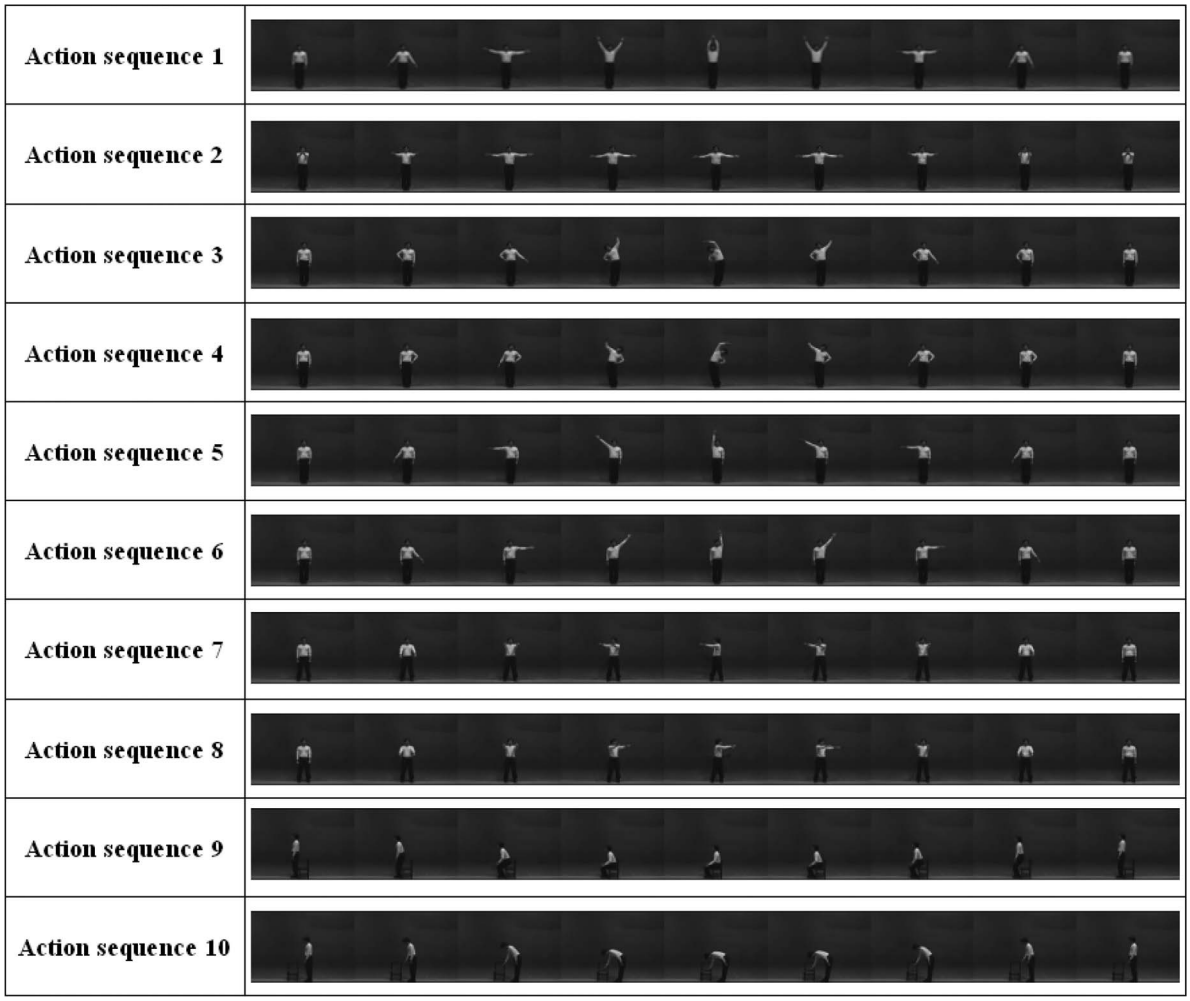


Fig. 8. Ten categories of atomic actions used for training.

improvement in computation time and little sacrifice in accuracy is basically tolerable. In addition, in this special application domain, the loss in accuracy can be easily compensated by checking the context information provided by the VLMM.

The average log likelihood of the training error computed with the training data is shown in Table I. The results indicate that the proposed action recognition method can deal with the problem of human action recognition effectively. Next, a test video was used to assess the effectiveness of the proposed method. The test data were obtained from the same human subject. Each atomic action was repeated four times, yielding a total of 40 test samples (4 positive samples and 36 negative samples) for evaluating the performance of the learned VLMMs. The proposed method achieved a 100% recognition rate for all the test sequences. To further verify the recognition results, we tested the similarity of any two VLMMs obtained in the experiment. First, we generated 10 000 action sequences for each of the ten VLMMs, which yielded a total of 100 000 action sequences. Out of the 100 000 action sequences, only 74 sequences were incorrectly recognized, and all the errors were on actions 7 and 8 because these two sequences contained many similar postures and thus could be mixed up easily (refer to Fig. 8). This result is consistent with the data shown in Table I: The log likelihood of actions 7 and 8 computed using VLMMs 8 and 7 was relatively high. This result confirms that the data shown in Table I are valid. Furthermore, we have also estimated the p-values [29] for each action model. The posture templates shown in Fig. 9 were used to generate 10 000 random action sequences using a sample-with-replacement process. The histograms of the log likelihood of the random sequences and the positive sequences for an action model are shown in Fig. 10. Since these two histograms do not overlap at all, it is reasonable to infer that the p-value of the action model is very low. To estimate the p-value, we approximate the distributions of the log likelihood by Gaussian distributions (see Fig. 10). Therefore, the p-value can be easily computed. The maximum p-value of the ten models is smaller than 0.0001, which confirms that the results are statistically significant.

In the third experiment, test videos of nine different human subjects (see Fig. 11) were used to evaluate the performance of the proposed method. Each person repeated each action five times, so we had five sequences for each action and each human subject, which yielded a total of 450 action sequences. For comparison, we also tested the performance of the HMM method in this experiment. Since the ten atomic actions used in the experiments were acyclic, only the left-right HMMs were

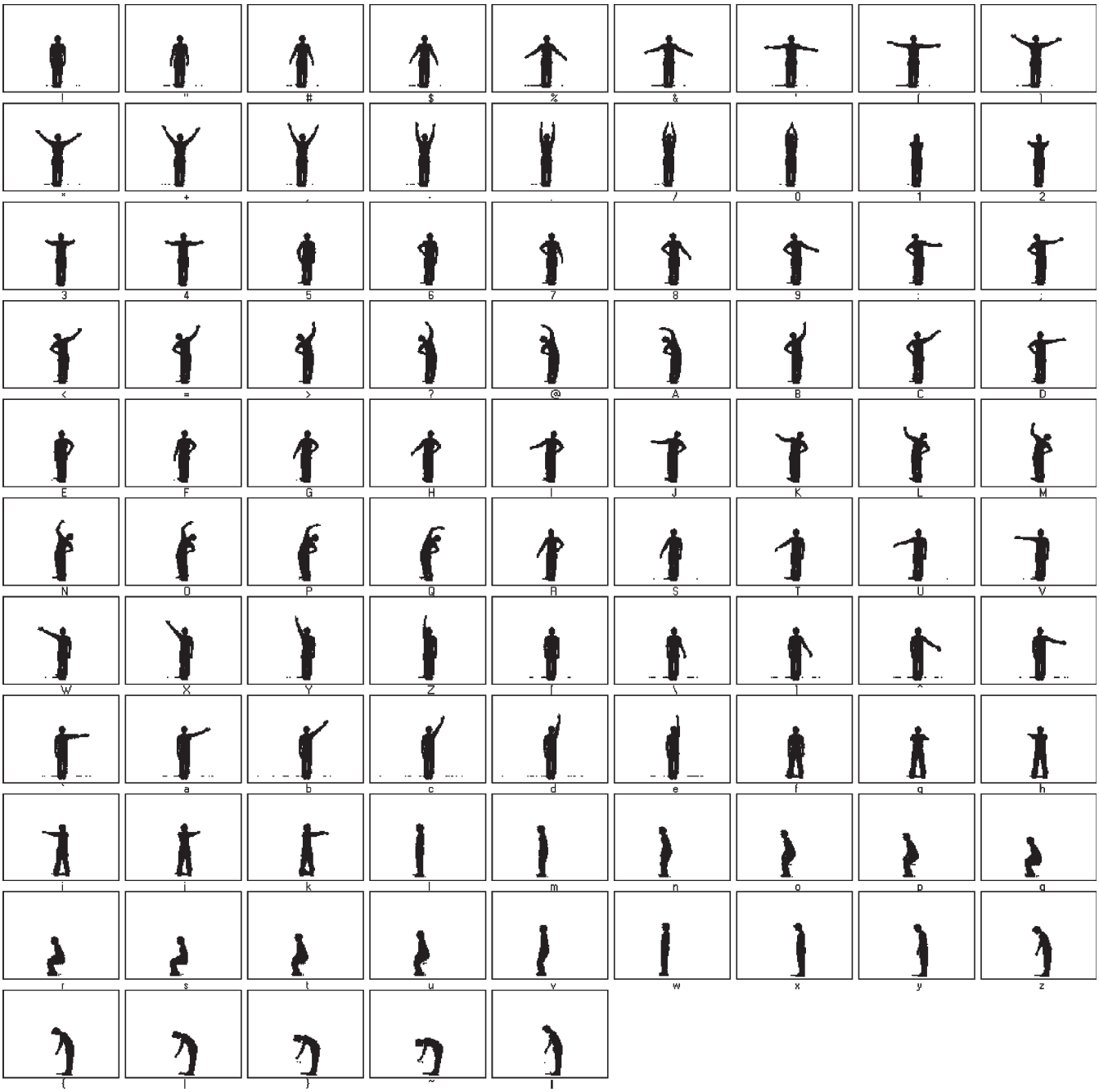


Fig. 9. Posture templates extracted from the training data.

considered in this experiment. Because the initial parameters and the number of HMM states would affect recognition results, the HMM implementation was evaluated using a variety of HMMs, each of which had a different number of hidden states. Furthermore, the HMM was trained ten times, and the average results were used to reduce the effect of the initial random parameters. Table II compares our method's recognition rate with that of the HMM method, for test data from nine different human subjects. Our method clearly outperforms the HMM method, no matter how many states were selected. In Table II, the shaded cells denote the best recognition results of the HMM approach for a particular action. It is clear that the selection of the number of states is a critical issue for the HMM method. Note that the number of HMM states that could be set for deriving the best performance was varying in different actions, which makes the selection of the number of states even more difficult. In contrast to the difficulty in determining the topology of an HMM, our method is simple and effective because the topology of a VLMM can be determined automatically with a robust algorithm. Note that the recognition rates for action 1 were the worst across all actions. Fig. 12(a) shows some typical input postures for a human subject performing action 1. The retrieved corresponding closest posture templates in the database are shown in Fig. 12(b). When comparing the corresponding posture templates shown in Fig. 12(b) with the training posture sequences shown in Fig. 8, it is clear that

TABLE I
RESULTS OF ATOMIC ACTION RECOGNITION USING THE TRAINING DATA

VLMM Log Likelihood / Action	1	2	3	4	5	6	7	8	9	10
1	-5.707	-63.78	-73.66	-81.2	-91.82	-91.12	-240	-211.1	-206.3	-239.9
2	-27.82	-5.944	-83.2	-67.84	-109.1	-107.7	-167.7	-156.1	-358.1	-259.5
3	-49.42	-76.62	-5.39	-64.79	-65.27	-42.27	-110.1	-100.6	-158.6	-162.6
4	-52.99	-83.22	-75.9	-5.524	-50.32	-80.6	-108.7	-115.6	-157.8	-177.5
5	-71.96	-81.93	-66.6	-46.16	-5.603	-89.7	-111.1	-119.1	-125.5	-119
6	-79.9	-100	-39.61	-87.91	-94.46	-5.559	-142.6	-126	-178.6	-254.2
7	-122.7	-75.95	-91.43	-110.7	-85.54	-96.43	-5.764	-9.797	-150.2	-150
8	-117.4	-87.62	-104.6	-103.9	-117.1	-81.35	-24.05	-5.884	-135.7	-159.5
9	-152.6	-149.3	-171.6	-131.4	-134.4	-124.3	-141.3	-140.9	-5.134	-111.8
10	-185.4	-198.1	-161.3	-166.7	-128.6	-224.1	-189.3	-192	-206.6	-5.453

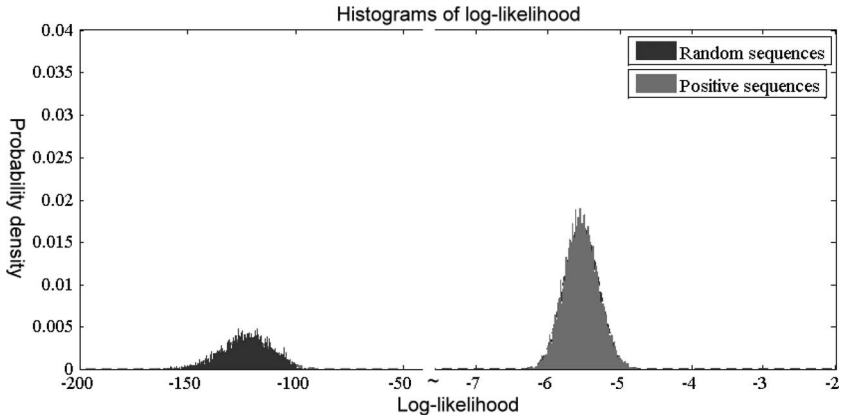


Fig. 10. Histograms of the log likelihood of the random sequences and the positive sequences for an action model.

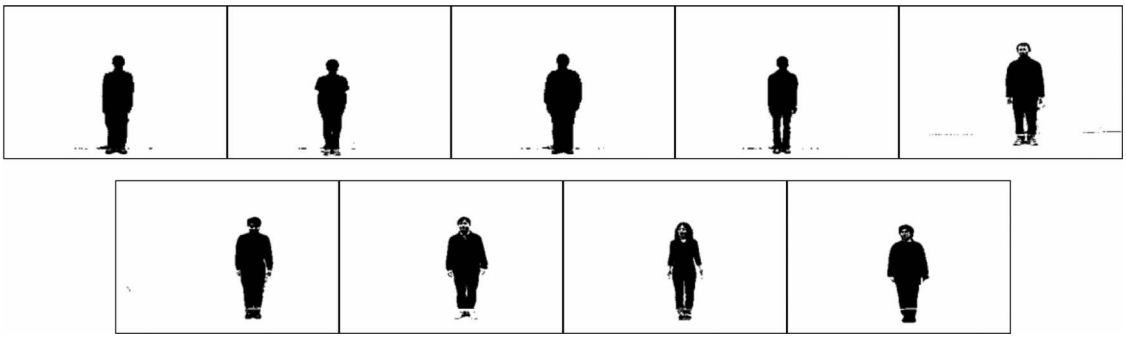


Fig. 11. Nine test human subjects.

the posture templates and the training postures of action 1, in this case, are not well matched. Due to the segmentation error of the lower arm areas, the input postures were incorrectly related to posture templates of different actions. For example, the retrieved posture templates shown in Fig. 12(b), from left to right, were extracted from training data of actions 1, 4, 2, 2, 2, 1, 2, 2, 2, 4, and 1, respectively. Since the proposed method is silhouette based, when the same postures of two individuals appear to be drastically different (due to dissimilar physical characteristics, motion styles, or improper segmentation), observation errors would bias the recognition result. In particular, if most of the input postures are with high observation error, the context information is not sufficient for accurate performance.

In order to show that the selection of the parameter τ_c in the posture template selection process was not a major concern, we calculated the recognition rates for different τ_c's. Fig. 13 shows the recognition rates with respect to different τ_c's, and it demonstrates that the change of τ_c only has little influence on the recognition results.

In the fourth experiment, to evaluate the scalability of the proposed algorithm, we used a new publicly available

TABLE II
COMPARISON OF OUR METHOD'S RECOGNITION RATE WITH THAT OF THE HMM COMPUTED WITH THE TEST DATA OBTAINED FROM NINE DIFFERENT HUMAN SUBJECTS

Actions / Recognition rate(%) / Methods	1	2	3	4	5	6	7	8	9	10
Our method	88.89	97.78	100	100	100	100	97.78	100	100	97.78
HMM (5 states)	88.22	82.00	93.78	87.78	88.22	90.89	96.89	99.78	90.44	97.56
HMM (10 states)	87.56	78.89	93.11	92.00	97.78	77.78	96.22	98.44	87.33	97.78
HMM (15 states)	88.89	81.33	93.11	93.11	92.89	66.44	97.33	99.33	98.89	97.33
HMM (20 states)	88.89	80.00	93.33	92.22	95.56	77.56	97.11	98.44	90.89	97.56
HMM (25 states)	88.89	81.56	93.56	92.67	93.56	60.89	95.56	100	98.89	97.56
HMM (30 states)	88.89	81.33	93.78	93.56	94.00	57.78	95.56	99.78	85.78	97.33

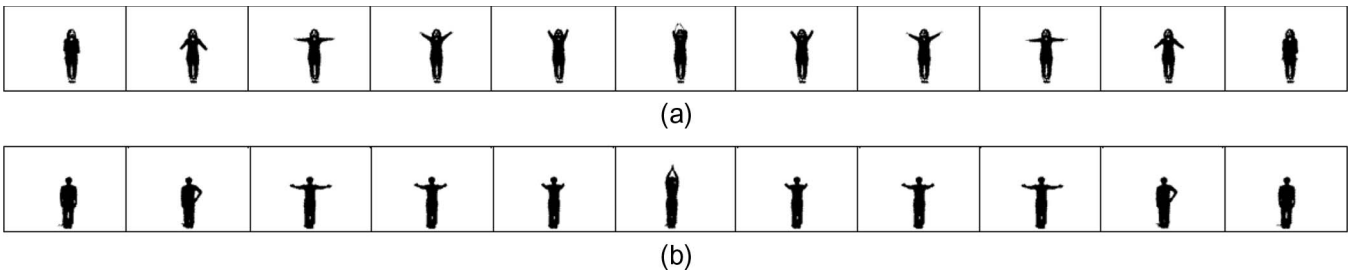


Fig. 12. Some typical postures of a human subject exercising action 1. (a) Input posture sequence. (b) Corresponding minimum-CSC-distance posture templates.

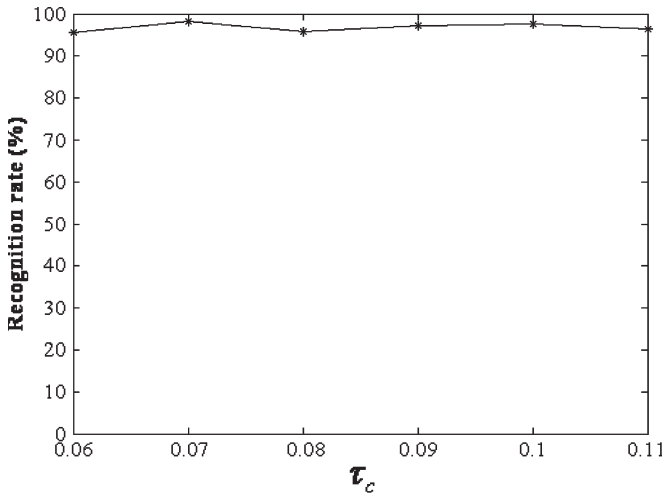


Fig. 13. Recognition rates with respect to different τ_c's.

database [4], [25]. This database consists of 90 low-resolution (180×144) action sequences from nine different people, each performing ten natural actions. These actions include bending (bend), jumping jacks (jack), jumping forward on two legs (jump), jumping in place on two legs (pjump), running (run), galloping sideways (side), skipping (skip), walking (walk), waving one hand (wave1), and waving two hands (wave2). Sample images of each type of action sequence are shown in Fig. 14. In [25], a sequence of human silhouettes derived from each action sequence was converted into two representations, namely, average motion energy (AME) and mean motion shape (MMS). Subsequently, a nearest neighbor classifier (NN) was used for recognition, and the leave-one-out cross-validation rule was adopted to compute the recognition rate. Recognition results for these two representations, shown in the top two rows of Table III, are compared against our method.

In order to compare our method with the two competing methods in a fairer fashion, we also applied the leave-one-out rule to our method. In this case, eight sets of data grabbed from eight distinct human subjects were used to train the VLMMs, resulting in eight VLMMs for each action. Finally, the category with the maximum likelihood was deemed to be the recognition result. Results using this methodology are shown in the last row

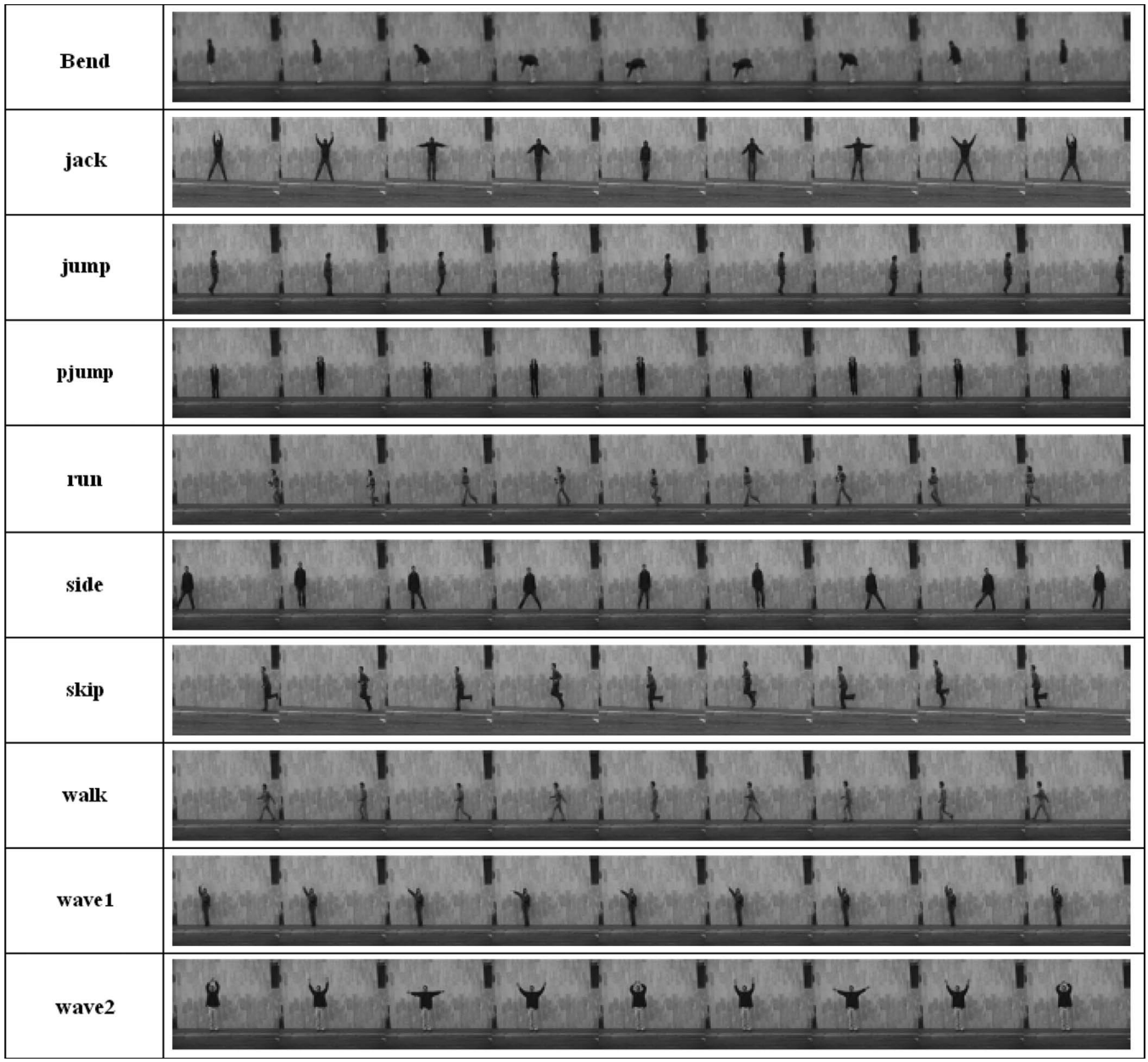


Fig. 14. Sample images in the public action database.

TABLE III
COMPARISON OF OUR METHOD'S RECOGNITION RATE WITH THAT OF THE AME PLUS NN METHOD AND THE MMS PLUS NN METHOD FOR THE PUBLIC DATABASE

Actions / Recognition rate(%) / Methods	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2	total
AME plus NN [25]	100	100	88.89	100	100	100	88.89	100	88.89	100	96.67
MMS plus NN [25]	100	100	88.89	77.78	100	88.89	100	88.89	88.89	100	93.33
Our method	100	100	100	100	100	100	100	100	100	100	100

of Table III. It is clear that our method outperforms the other two methods for this public database.

V. Conclusion

We have proposed a framework for understanding human atomic actions using VLMMs. The framework comprises the following two modules: a posture labeling module and a VLMM atomic action learning and recognition module. We have developed a simple and efficient posture template selection algorithm based on a modified shape context matching method. A codebook of posture templates is created to convert the input posture sequences into discrete symbols so that the language modeling approach can be applied. The VLMM technique is then used to learn human action sequences. To handle the dynamic time warping problem and the lack of noise observation model problem of applying the VLMM technique to behavior analysis, we have also developed a systematic method to convert the learned VLMMs into HMMs. The contribution of our approach is that the topology of the HMMs can be automatically determined and that the recognition accuracy is better than the traditional HMM approach. Experimental results demonstrate the efficacy of the proposed method.

References

[1] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Comput. Vis. Image Understanding*, vol. 73, no. 3, pp. 428–440, Mar. 1999.

[2] H. A. Baler Saip and C. L. Lucchesi, "Matching algorithm for bipartite graph," Departamento de Cincia da Computao, Universidade Estudal de Campinas, Campinas, Brazil, Tech. Rep. DCC-03/93, 1993.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1395–1402.

[5] A. F. Bobick and Y. A. Ivanov, "Action recognition using probabilistic parsing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Santa Barbara, CA, 1998, pp. 196–202.

[6] D. Y. Chen, S. W. Shih, and H.-Y. M. Liao, "Atomic human action segmentation using a spatio-temporal probabilistic framework," in *Proc. IEEE Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Pasadena, CA, 2006, pp. 327–330.

[7] R. T. Collins, A. J. Lipton, and T. Kanade, "Introduction to the special section on video surveillance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 745–746, Aug. 2000.

[8] W. B. Croft and J. Lafferty, *Language Modeling for Information Retrieval*. Norwell, MA: Kluwer, 2003.

[9] A. Galata, N. Johnson, and D. Hogg, "Learning variable-length Markov models of behavior," *Comput. Vis. Image Understanding*, vol. 81, no. 3, pp. 398–413, Mar. 2001.

[10] D. M. Gavrila, "The visual analysis of human movement: A survey," *Comput. Vis. Image Understanding*, vol. 73, no. 1, pp. 82–98, Jan. 1999.

[11] P. Guttorp, *Stochastic Modeling of Scientific Data*. London, U.K.: Chapman & Hall, 1995.

[12] I. Guyon and F. Pereira, "Design of a linguistic postprocessor using variable memory length Markov models," in *Proc. Int. Conf. Document Anal. Recog.*, Montréal, QC, Canada, 1995, pp. 454–457.

[13] J. W. Hsieh, Y. T. Hsu, H.-Y. M. Liao, and C. C. Chen, "Video-based human movement analysis and its application to surveillance systems," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 372–384, Apr. 2008.

[14] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press, 1998.

[15] H. Miyamori and S. Iisaku, "Video annotation for content-based retrieval using human behavior analysis and domain knowledge," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recog.*, Grenoble, France, 2000, pp. 320–325.

[16] A. S. Ogale, A. Karapurkar, and Y. Aloimonos, "View-invariant modeling and recognition of human actions using grammars," in *Proc. Workshop Dynamical Vis. ICCV*, Beijing, China, 2005, pp. 115–126.

[17] J. Park, S. Park, and J. K. Aggarwal, "Model-based human motion tracking and behavior recognition using hierarchical finite state automata," in *Proc. Int. Conf. Comput. Sci. Appl.*, Assisi, Italy, 2004, pp. 311–320.

[18] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[19] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia," in *Advances in Neural Information Processing Systems*. New York: Morgan Kaufmann, 1994, pp. 176–183.

[20] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proc. IEEE*, vol. 88, no. 8, pp. 1270–1278, Aug. 2000.

[21] R. Sharma, V. I. Pavlović, and T. S. Huang, "Toward multimodal human–computer interface," *Proc. IEEE*, vol. 86, no. 5, pp. 853–869, May 1998.

[22] C. W. Su, H.-Y. M. Liao, H. R. Tyan, C. W. Lin, D. Y. Chen, and K. C. Fan, "Motion flow-based video retrieval," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1193–1201, Oct. 2007.

[23] A. Vinciarelli, S. Bengio, and H. Bunke, "Offline recognition of unconstrained handwritten texts using HMMs and statistical language models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 709–720, Jun. 2004.

[24] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognit.*, vol. 36, no. 3, pp. 585–601, Mar. 2003.

[25] L. Wang and D. Suter, "Informative shape representations for human action recognition," in *Proc. IEEE Int. Conf. Pattern Recog.*, 2006, vol. 2, pp. 1266–1269.

[26] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[27] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1992, pp. 379–385.

[28] J. Yang, Y. Xu, and C. S. Chen, "Human action learning via hidden Markov model," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 27, no. 1, pp. 34–44, Jan. 1997.

[29] [Online]. Available: http://en.wikipedia.org/wiki/Statistical_significance

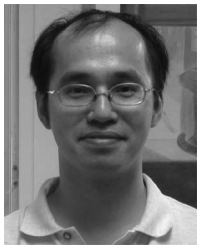
[30] [Online]. Available:http://www.iis.sinica.edu.tw/~ulin/Behavior

Yu-Ming Liang received the B.S. and M.S. degrees in information and computer education from National Taiwan Normal University, Taipei, Taiwan, in 1999 and 2002, respectively. He has been working toward the Ph.D. degree in the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, since 2004.

His research interests include computer vision, pattern recognition, and multimedia signal processing.

Sheng-Wen Shih received the M.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1990 and 1996, respectively.

From January to July 1997, he was a Postdoctoral Associate with the Image Formation and Processing Laboratory, Beckman Institute, University of Illinois, Urbana. He has been an Associate Professor with the Department of Computer Science and Information Engineering, National Chi Nan University, Nantou, Taiwan, since 2003, where he was an Assistant Professor in August 1997. His research interests include computer vision, biometrics, and human–computer interaction.



Arthur Chun-Chieh Shih was born in Taipei, Taiwan, on September 26, 1966. He received the B.S. degree in electrical engineering from Chinese Culture University, Taipei, in 1992, the M.S. degree in electrical engineering from National Chung Cheng University, Chiayi, Taiwan, in 1994, and the Ph.D. degree in computer science and information engineering from National Central University, Chung-Li, Taiwan, in 1998.

From October 1998 to July 2002, he was with the Institute of Information Science, Academia Sinica, Taipei, and the Department of Ecology and Evolution, University of Chicago, Chicago, IL, as a Postdoctoral Fellow. He has been an Associate Research Fellow with the Institute of Information Science, Academia Sinica, Taipei, since 2008, where he was an Assistant Research Fellow in July 2002. His current research interests include molecular evolution, bioinformatics, and multimedia signal processing.

Hong-Yuan Mark Liao received the B.S. degree in physics from National Tsing Hua University, Hsinchu, Taiwan, in 1981 and the M.S. and Ph.D. degrees in electrical engineering from Northwestern University, Evanston, IL, in 1985 and 1990, respectively.

He was a Research Associate with the Computer Vision and Image Processing Laboratory, Northwestern University, during 1990–1991. In July 1991, he was with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, as an Assistant Research Fellow. He was promoted to Associate Research Fellow and then Research Fellow in 1995 and 1998, respectively. From August 1997 to July 2000, he was the Deputy Director of the institute. From February 2001 to January 2004, he was the Acting Director of the Institute of Applied Science and Engineering Research. He is jointly appointed as a Professor with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu. He is the Editor-in-Chief of the *Journal of Information Science and Engineering*. He is also an Editorial Board Member of the *International Journal of Visual Communication and Image Representation*, the *EURASIP Journal on Advances in Signal Processing*, and the *Research Letters in Signal Processing*. His current research interests include multimedia signal processing, video-based surveillance systems, content-based multimedia retrieval, and multimedia protection.

Dr. Liao was the recipient of the Young Investigators' Award from Academia Sinica in 1998, the Excellent Paper Award from the Image Processing and Pattern Recognition Society of Taiwan in 1998 and 2000, the Distinguished Research Award from the National Science Council of Taiwan in 2003, the National Invention Award of Taiwan in 2004, and the Distinguished Scholar Research Award from the National Science Council of Taiwan in 2008. He was an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA during 1998–2001. He served as the Program Cochair of the Second IEEE Pacific Rim Conference on Multimedia. He was the Conference Cochair of the Fifth International Conference on Multimedia and Exposition (ICME 2004) in June 2004, the Technical Cochair of ICME 2007, a committee member of the 2005, 2006, and 2007 ACM Multimedia Conferences and the 2007 World Wide Web Conference.

Cheng-Chung Lin received the B.S. degree in control engineering and the M.S. degree in computer engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1979 and 1985, respectively, and the Ph.D. degree from the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, in 1997.

He was a Digital Circuits/Systems Design Engineer with the R&D Department, TECO Inc., Taiwan, during 1979 and 1981 and was then with NCTU until 1990. He is currently an Associate Professor with the Department of Computer Science, NCTU. His research interests include computer architectures, computer graphics, computer vision, and face recognition.