中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-07-008

A Language Modeling Approach to Atomic Human Action Recognition

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

April 16, 2007 || Technical Report No. TR-IIS-07-008
http://www.iis.sinica.edu.tw/LIB/TechReport/tr2007/tr07.html

A Language Modeling Approach to Atomic Human Action Recognition

Yu-Ming Liang[a], Sheng-Wen Shih[b], Arthur Chun-Chieh Shih[c], Hong-Yuan Mark Liao[a,c], and Cheng-Chung Lin[a]

[a]Department of Computer Science,
National Chiao Tung University, Hsinchu 300, Taiwan
ulin@iis.sinica.edu.tw

[b]Department of Computer Science and Information Engineering,
National Chi Nan University, Puli, Nantou 545, Taiwan
swshih@ncnu.edu.tw

[c]Institute of Information Science,
Academia Sinica, Nankang, Taipei 115, Taiwan
{arthur, liao}@iis.sinica.edu.tw

Abstract

Visual analysis of human behavior has generated considerable interest in the field of computer vision because it has a wide spectrum of potential applications. A human behavior analysis system must address three main tasks: object detection, human tracking, and understanding behavior. In this paper, we propose a language modeling framework for handling the third task. The framework is comprised of two modules: a posture labeling module, and an atomic action learning and recognition module. A posture template selection algorithm is developed based on a modified shape context matching technique. The posture templates form a codebook that is used to convert input posture sequences into training symbol sequences or recognition symbol sequences. Finally, a variable-length Markov model technique is applied to learn and recognize the input symbol sequences of atomic actions. Experiments on real data demonstrate the efficacy of the proposed system.

1. Introduction

In recent years, visual analysis of human behavior has generated considerable interest in the field of computer vision because it has a wide spectrum of potential applications, such as smart surveillance [5], human computer interfaces [18], content-based retrieval [12], and virtual reality [21]. Comprehensive surveys of related work can be found in [1, 8, 20]. 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 [20]. While the low-level processes have been studied extensively, the high-level process has received less attention. 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 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, which indicates a unitary and complete human movement, 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 behaviors involves 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 [4]. 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 [11, 17], a powerful tool for dealing with temporal ordering problems, has been applied in many fields, such as speech recognition [11], handwriting recognition [19], and information retrieval [6]. 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 [3] and Ogale et al. [13] used context-free grammars to model human actions, while Park et al. employed hierarchical finite state automata to recognize human behavior [14]. In [22, 23], hidden Markov models (HMM) were applied to human action recognition. This particular language modeling technique is useful for both human action recognition and human action sequence synthesis. Galata et al. utilized variable-length Markov models (VLMM) to characterize human actions [7], 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 can be categorized into two classes: deterministic algorithms [3, 13, 14] and stochastic algorithms [7, 22, 23]. 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 a HMM are not observable, encoding high-order temporal dependencies with this model is a challenging task. There is no systematic way to determine the topology of a 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-term and short-term dependencies 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 two limitations: 1) it cannot handle the dynamic time warping problem, and 2) it lacks a model for observing noise.

In this research, we propose a hybrid framework of VLMM and HMM that retains the models advantages, while avoiding their drawbacks. The framework is comprised of three modules: a posture labeling module, a VLMM atomic action learning module, and a 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 a 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 a HMM.

The remainder of this paper is organized as follows. In Section 2, we introduce the theory of VLMM. The proposed approach is described in Section 3, and the experiment results are detailed in Section 4. Then, in Section 5, we present our conclusions.

2. Variable Length Markov Model

A variable length Markov model technique [7, 10, 16] is frequently applied to language modeling problems because of its powerful ability to encode temporal dependencies. As shown in Fig. 1, a VLMM can be regarded as a probabilistic finite state automaton (PFSA) $\Lambda = (S, V, \tau, \gamma, \pi)$ [16], where

- 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,
- V denotes a finite observation alphabet,
- $\tau : S \times V \rightarrow S$ is a state transition function such that $\tau(s_i, v) \rightarrow s_j$,
- $\gamma : S \times V \rightarrow [0,1]$ represents the output probability function with $\forall s \in S, \sum_{v \in V} \gamma(s, v) = 1,$ and
- $\pi : S \rightarrow [0,1]$ is the probability function of the initial state that will satisfy $\sum_{s \in S} \pi(s) = 1$.

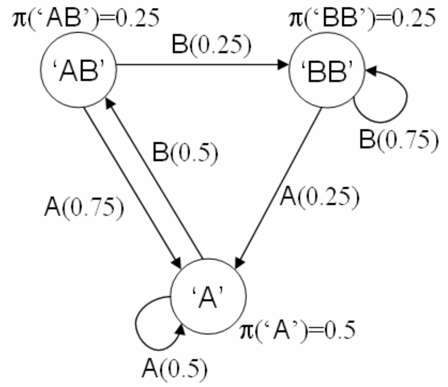


Figure 1. An example of a VLMM

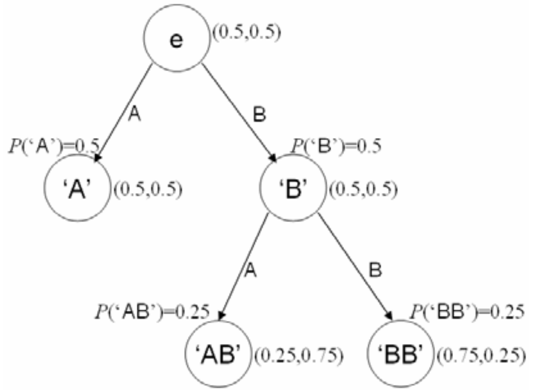


Figure 2. The PST for constructing the PFSA shown in Figure 1

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 constructing a PFSA involves constructing a prediction suffix tree (PST) [16]. A prediction suffix tree *T* over *V* is a tree of degree |*V*|. Figure 2 depicts the PST constructed from an observation sequence generated by the VLMM shown in Fig. 1. Except for the root node, each node of the PST represents a non-empty symbol string, and each parent node represents the longest suffix tree of its child nodes. In addition, $P(v \mid s)$ is the output probability distribution of the next symbol v of each node *s* that satisfies $\sum_{v\in V} P(v \mid s) = 1$. The output and prior probabilities can be derived from the training symbol sequences as follows

$$P(v \mid s) = \frac{N(sv)}{N(s)}, \tag{1}$$

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

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 $P(v \mid s)$, and $v's$ is its child node with the output probability $P(v \mid v's)$. 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 $P(v \mid v's)$ used to predict the next symbol, *v*, is significantly better than the output probability $P(v \mid s)$ 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-Lievler (KL) divergence is applied to measure the statistical difference between the probabilities $P(v \mid v's)$ and $P(v \mid s)$ as follows:

$$\Delta H(v's, s) = P(v's) \sum_{v} P(v \mid v's) \log \frac{P(v \mid v's)}{P(v \mid s)}. \tag{3}$$

If $\Delta H(v's, s)$ is greater than a given threshold, the node $v's$ 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 Figures 1 and 2. The procedure for transforming a PST into its corresponding VLMM is described in [16]. 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, $O = o_1 o_2 \ldots o_T$, into the product of conditional probabilities as follows:

$$P(O \mid \Lambda) = \prod_{j=1}^{T} P(o_j \mid o_{j-d_j} \cdots o_{j-1}, \Lambda), \tag{4}$$

where o_j is the *j*-th 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, $O = o_1 o_2 ... o_T$, in terms of the highest probability. Therefore, the recognition result can be determined as model i^* as follows:

$$i^* = \arg\max_i P(O \mid \Lambda_i). \tag{5}$$

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 1, 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 observing noise, so the system is less tolerant of image preprocessing errors. We describe our solutions to these two problems in the next section.

3. The Proposed Method for Atomic Action Recognition

The proposed method comprises 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 and recognize the constructed symbol sequences. The two phases are described below.

3.1 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. Figure 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 subsections.

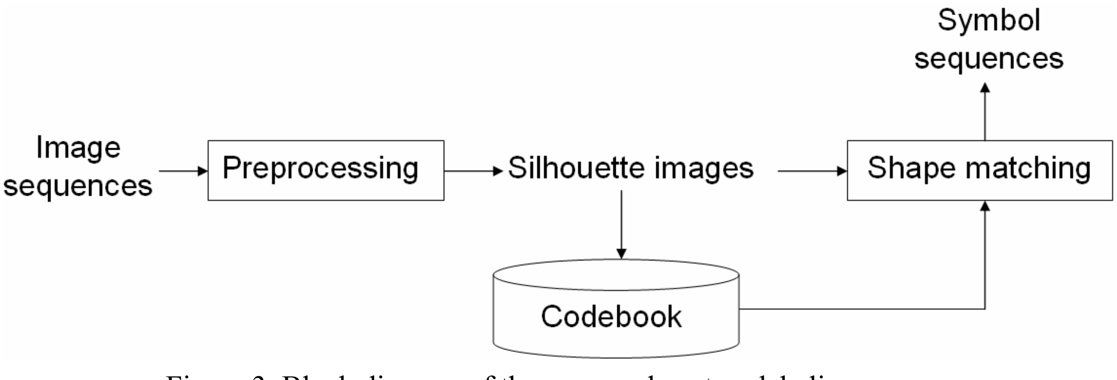


Figure 3. Block diagram of the proposed posture labeling process

3.1.1 Shape matching with a modified shape context technique

We modified the shape context technique proposed by Belongie et al. [2] to deal with the shape matching problem. In the original shape context approach, a shape is represented by a discrete set of sampled points, $P = \{p_1, p_2, ..., p_n\}$. 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.

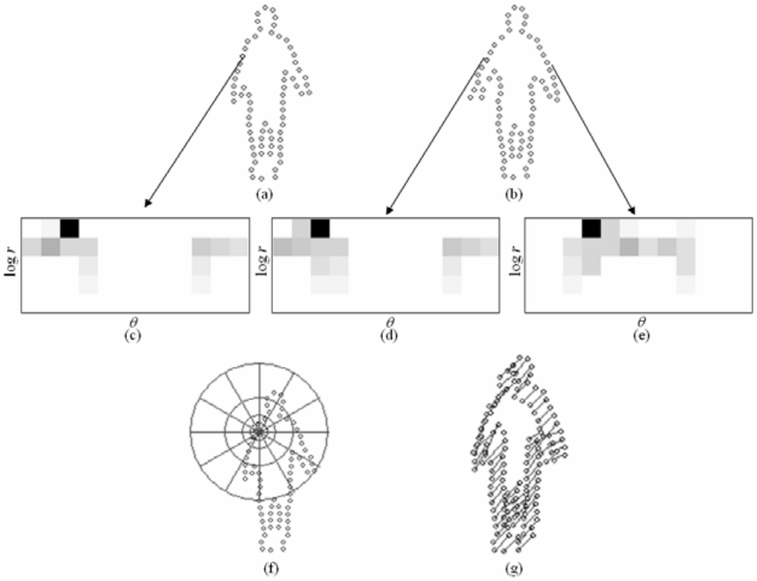


Figure 4. Shape context computation and matching: (a) and (b) show the sampled points of two shapes; and (c)-(e) are the local shape contexts corresponding to different reference points. A diagram of the log-polar space is shown in (f), while (g) shows the correspondence between points computed using 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:

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

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.

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 noise, 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. Shape matching based on this modified shape context technique is accomplished by minimizing the total cost of the matching modified in Equation (7) as follows:

$$H'(\pi) = \sum_{j \in A} C(p_j, q_{\pi(j)}), \tag{8}$$

where A is the set of convex hull vertices. An example of convex hull-shape contexts matching is shown in Fig. 5. There are three important reasons why *convex hull-shape contexts* (CSC) 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.

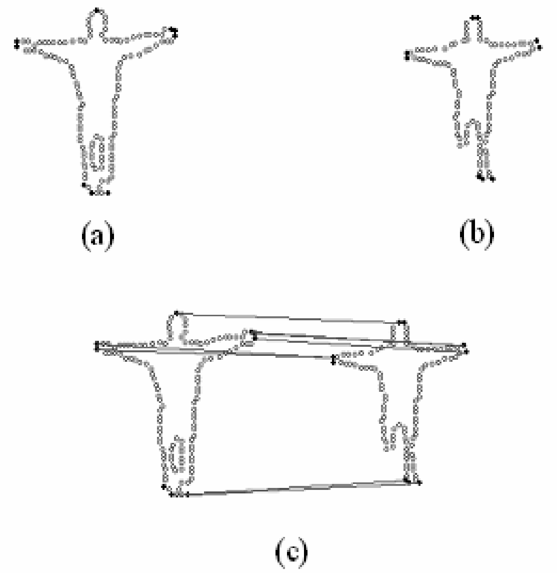


Figure 5. Convex hull-shape contexts matching: (a) and (b) show the convex hull vertices of two shapes; (c) shows the correspondence between the convex hull vertices determined using shape matching.

3.1.2 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 3.1.1. In the method, the cost of matching two shapes, see Equation (8), is denoted by $C_{cp}(b_i, a_j)$. 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.

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

3.2 Human action sequence learning and recognition

Using the posture templates codebook, an input sequence of postures $\{b_1, b_2, ..., b_n\}$ can be converted into a symbol sequence $\{a_{q(1)}, ..., a_{q(n)}\}$, where $q(i) = \arg \min_{j \in \{1,2,...,M\}} C_{cp}(b_i, a_j)$. Thus, atomic action VLMMs can be trained by the method outlined in Section 2. 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_i-th order Markov chain with state space *S* is given by

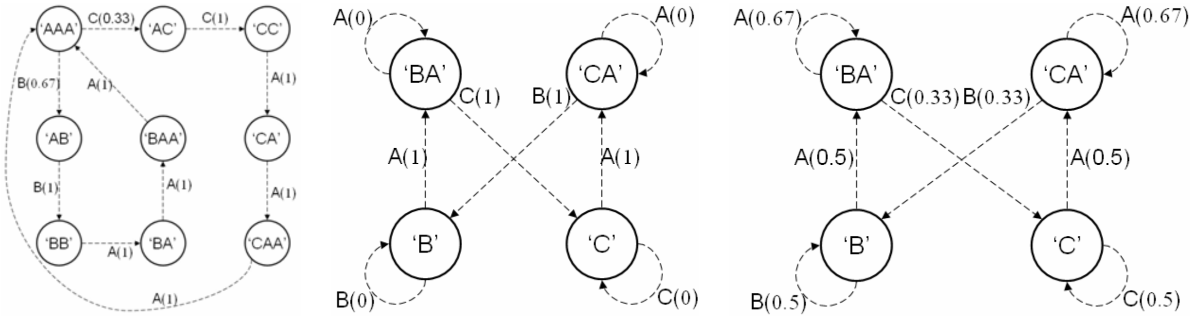
$$P(X_i = r_i \mid X_{i-d_i} = r_{i-d_i}, X_{i-d_i+1} = r_{i-d_i+1}, \cdots, X_{i-1} = r_{i-1}), \tag{9}$$

where X_i is a state in *S*. To transform the d_i-th order Markov chain into a first-order Markov chain, a new state space is constructed such that both $Y_{i-1} = (X_{i-d_i}, \cdots, X_{i-1}) = (r_{i-d_i} \cdots r_{i-1})$ and $Y_i = (X_{i-d_{i+1}-1}, \cdots, X_i) = (r_{i-d_{i+1}-1}, \cdots, r_i)$ 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 [9]

$$\begin{aligned} &P(X_i = r_i \mid X_{i-d_i} = r_{i-d_i}, X_{i-d_i+1} = r_{i-d_i+1}, \cdots, X_{i-1} = r_{i-1}) \\ &\quad = P(Y_i = (r_{i-d_{i+1}+1} \cdots r_i) \mid Y_{i-1} = (r_{i-d_i} \cdots r_{i-1})). \end{aligned} \tag{10}$$

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

As mentioned in Section 2, two problems must be solved before the VLMM technique can be applied to the action recognition task, namely, the dynamic time warping problem and the lack of a model for observing noise. 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. [7]. 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 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, $a_{ii}^{old} = 0$ because repeated symbols are merged into one symbol. Then, Equation (1) is used to update the probability of self-transition, $a_{ii}^{new} = P(v_i \mid v_i) = \frac{N(v_i v_i)}{N(v_i)}$, and the other transition probability is updated as $a_{ij}^{new} = a_{ij}^{old}\left(1 - a_{ii}^{new}\right)$. 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 modified VLMM constructed with the preprocessed training sequence are shown in Figures. 6(b) and 6(c), respectively.



(a) (b) (c)

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

Next, a noise observation model is introduced to convert a VLMM into a HMM. Note that the output of a VLMM determines its state transition and vice versa because the states of a VLMM are observable. However, due to the image preprocessing noise, the symbol sequence corresponding to an atomic action includes some randomness. If, according to the VLMM, the output symbol is q_t at time *t*, then its posture template a_t can be retrieved from the codebook. The extracted silhouette image o_t will not deviate too much from its corresponding posture template a_t if the segmentation result does not contain any major errors. Therefore, the CSC distance $C_{cp}\left(o_t, a_t\right)$ between the image and the template will be close to zero. In this work, we assume that the CSC distance has a Gaussian distribution, i.e., $P(o_t \mid q_t, \Lambda) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{C_{cp}(o_t, a_t)}{2\sigma^2}}$.

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 the state, then the VLMM becomes a standard HMM. The probability of the observed string of symbols, $O = o_1 o_2 ... o_T$, for a given model Λ can be evaluated by the HMM forward/backward procedure with proper scaling [15]. Finally, the category i^* that maximizes the following equation is deemed to be the recognition result:

$$i^* = \arg\max_i \log[P(O \mid \Lambda_i)] \,. \tag{11}$$

4. Experiments

We conducted a series o f experiments to evaluate the effectiveness of the proposed method. The training data used in the experiments was a real video sequence comprised of approximately 900 frames with ten categories of action sequences. Some typical image frames are shown in Fig. 7. Using the posture template selection algorithm, a codebook of 75 posture templates (see Fig. 8), was constructed from the training data. The data was then used to build ten VLMMs, each of which was associated with one of the atomic actions shown in Fig. 7.

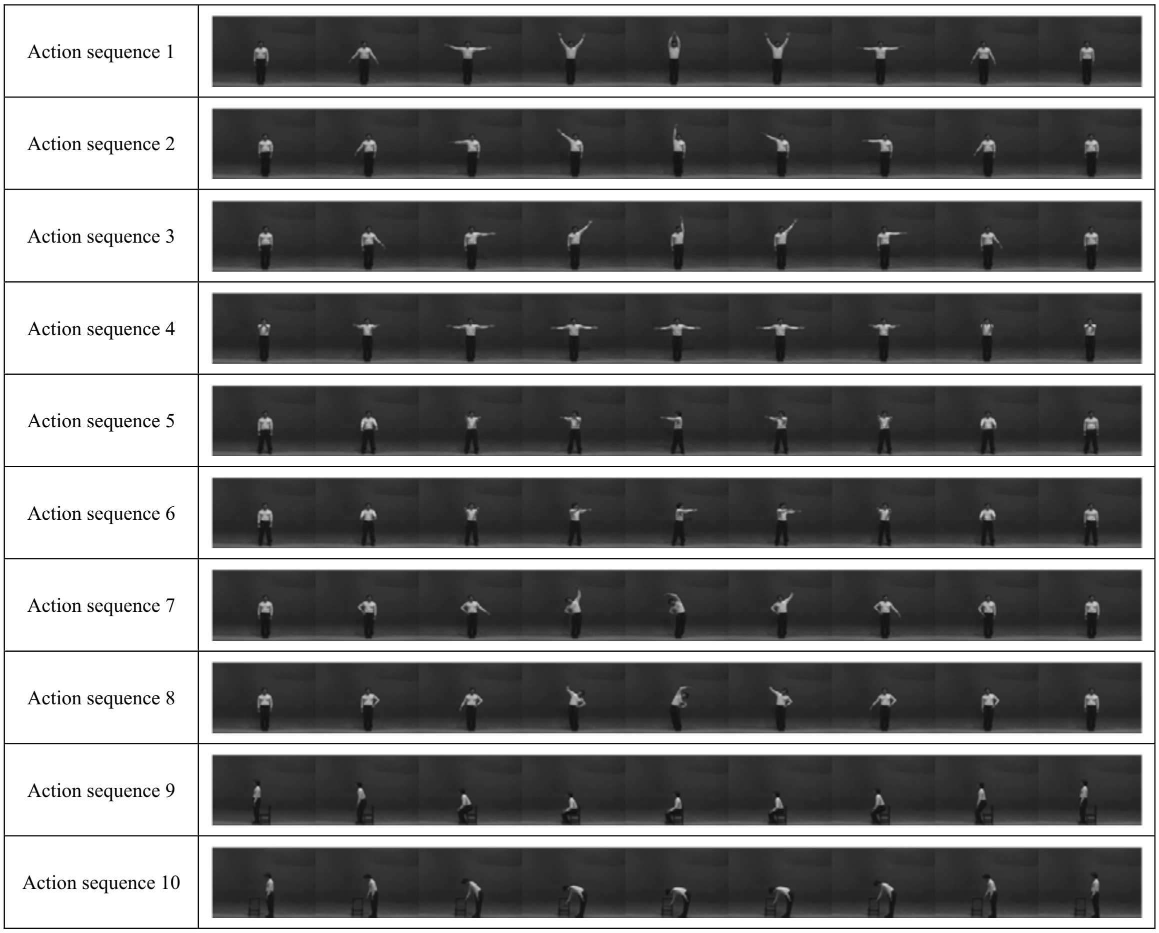


Figure 7. The ten categories of atomic actions used for training

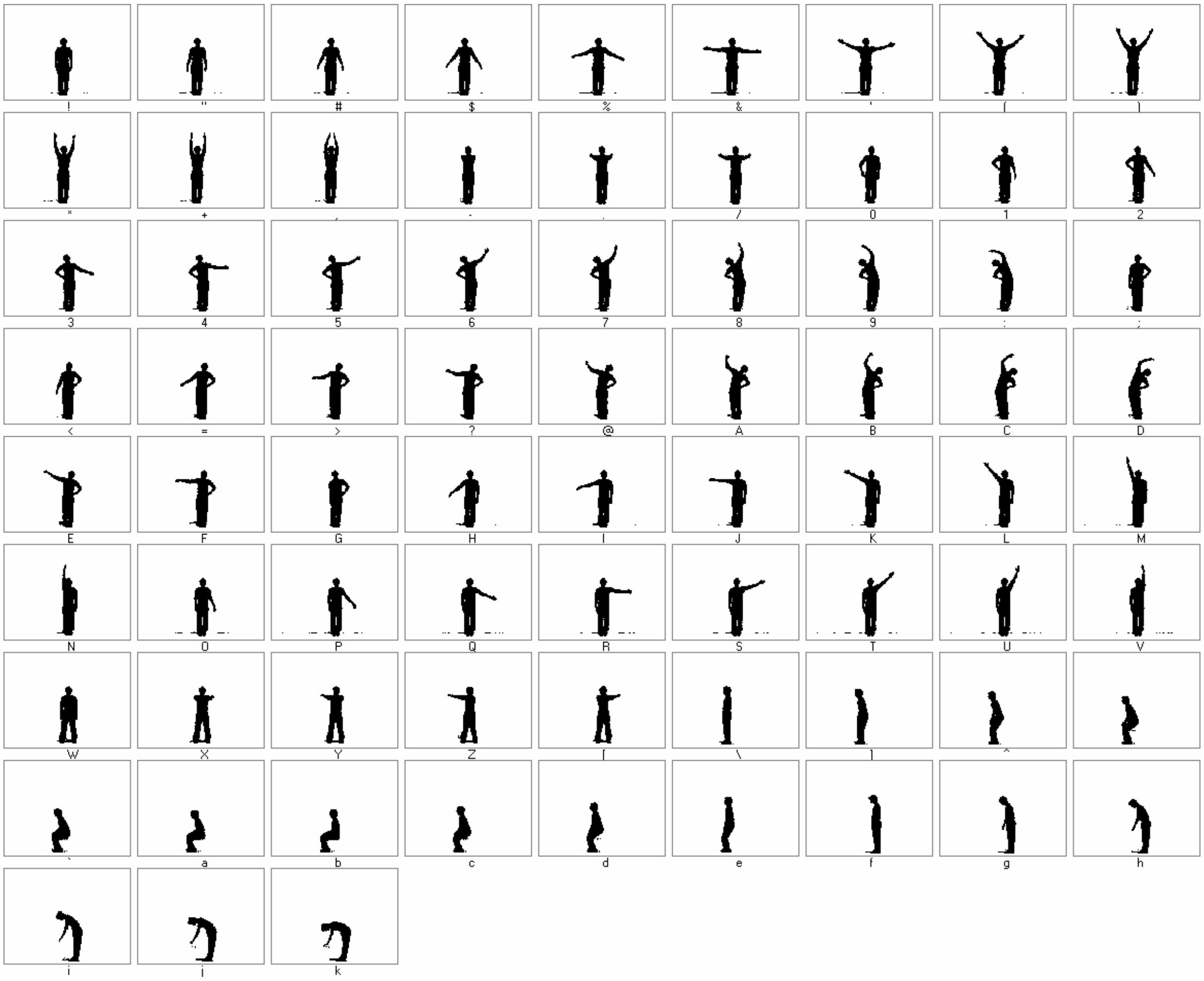


Figure 8. Posture templates extracted from the training data

The average log-likelihood of the training error computed with the training data is shown in Table 1. 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 was obtained from the same subject. Each atomic action was repeated four times, yielding a total of 40 action sequences. The proposed method achieved a 100% recognition rate for all the test sequences.

Table 1. The results of atomic action recognition using the training data

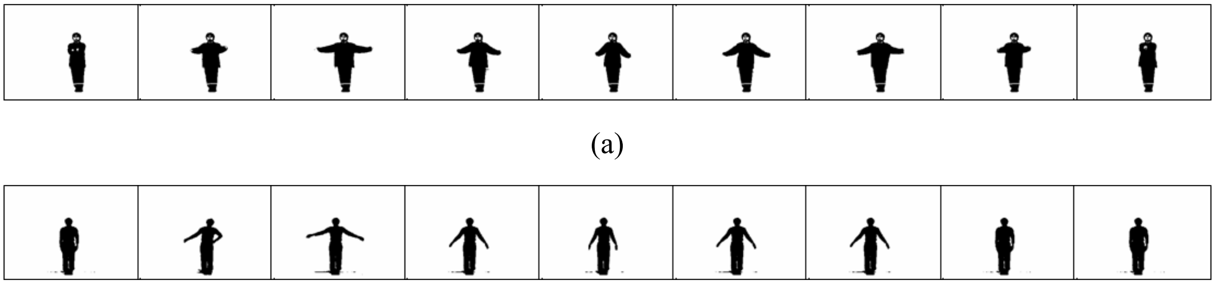
VLMM Recognition rate (%) Action	1	2	3	4	5	6	7	8	9	10
1	-4.068	-5.612	-5.665	-5.889	-5.538	-5.346	-6.541	-6.599	-6.296	-6.412
2	-4.969	-3.85	-6.071	-5.954	-6.18	-6.012	-5.945	-6.101	-6.391	-6.535
3	-5.243	-5.905	-3.926	-5.532	-5.227	-5.123	-6.008	-6.055	-6.537	-6.269
4	-5.467	-5.894	-5.657	-3.959	-5.288	-5.528	-6.078	-6.115	-6.433	-6.454
5	-5.084	-5.927	-5.443	-5.19	-3.975	-5.59	-6.033	-6.165	-6.373	-6.1
6	-5.235	-5.997	-5.246	-5.83	-5.909	-3.98	-6.261	-6.232	-6.519	-6.589
7	-7.125	-6.467	-6.949	-6.956	-6.711	-6.928	-3.734	-4.072	-6.627	-6.605
8	-7.138	-6.463	-6.942	-6.716	-6.941	-6.664	-4.026	-3.734	-6.585	-6.603
9	-6.664	-6.948	-7.084	-6.937	-6.659	-6.906	-6.515	-6.478	-3.891	-5.775
10	-7.252	-7.345	-6.787	-7.424	-7.027	-7.547	-6.797	-6.804	-6.331	-3.842

In the second experiment, test videos of nine subjects (see Fig. 9) 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 subject, which yielded a total of 450 action sequences. For comparison, we also tested the performance of the HMM method in this experiment. The HMMs we used were fully connected models. The number of states for each HMM was assigned as the number of states of the corresponding learned VLMM. Table 2 compares our method's recognition rate with that of the HMM method computed with the test data from the nine subjects. Our method clearly outperforms the HMM method. Note that the recognition rate of action 4 is the worst of the group. Figure 10(a) shows some typical incorrectly-labeled input postures of a subject exercising action 4. The computed incorrect posture templates are shown in Fig. 10(b). Comparison of the posture template sequences of the incorrectly recognized action with those of actions 1 and 2 in Fig. 7 shows the cause of the recognition error: the way this subject exercise action 4 are different from the training sample. To cope with this subject-dependent motion pattern discrepancy, we will have to use more training data of different subjects to train the recognition system to improve the recognition rate.

Figure 9. Nine test subjects

Table 2. Comparison of our method's recognition rate with that of the HMM computed with the test data from the nine subjects

Actions	1	2	3	4	5	6	7	8	9	10
Our method	88.89	100	100	84.44	100	100	97.78	100	100	100
HMM	88.89	86.67	100	77.78	97.78	100	88.89	95.56	100	100



(a)

(b)

Figure 10. One of the sequences has been recognized incorrectly: (a) some typical incorrectly labeled postures, (b) the corresponding incorrect posture templates

5. Conclusion

We have proposed a framework for understanding human atomic actions using a language modeling approach. The framework comprises 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 and recognize human action sequences. Our experiment results demonstrate the efficacy of the proposed system.

References

[1] J. K. Aggarwal and Q. Cai, “Human motion analysis: a review,” *Computer Vision and Image Understanding*, Vol. 73, No. 3, pp. 428- 440, 1999.

[2] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 24, pp. 509- 522, 2002.

[3] A. F. Bobick and Y. A. Invanov, “Action recognition using probabilistic parsing,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, pp. 192- 202, 1998.

[4] D. Y. Chen, S. W. Shih, and H. Y. Mark Liao, “Atomic human action segmentation using a spatio-temporal probabilistic framework,” Proceedings of IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Pasadena, CA, pp. 327- 330, 2006.

[5] R. T. Collins, A. J. Lipton, and T. Kanade, “Introduction to the special section on video surveillance,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 745- 746, 2000.

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

[7] A. Galata, N. Johnson, and D. Hogg, “Learning variable-length Markov models of behavior,” *Computer Vision and Image Understanding*,” Vol. 81, No. 3, pp. 398- 413, 2001.

[8] D. M. Gavrila, “The visual analysis of human movement: a survey,” *Computer Vision and Image Understanding*, Vol. 73, No. 1, pp. 82- 98, 1999.

[9] Peter Guttorp, *Stochastic Modeling of Scientific Data*, London: Chapman and Hall/CRC, 1995.

[10] I. Guyon and F. Pereira, “Design of a linguistic postprocessor using variable memory length Markov models,” *Proceedings of International Conference on Document Analysis and Recognition*, Montréal, Canada, pp. 454- 457, 1995.

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

[12] H. Miyamori and S. Iisaku, “Video annotation for content-based retrieval using human behavior analysis and domain knowledge,” *Proceedings of IEEE International Conference on*

Automatic Face and Gesture Recognition, Grenoble, France, pp. 320- 325, 2000.

[13] A. S. Ogale, A. Karapurkar, and Y. Aloimonos, “View-invariant modeling and recognition of human actions using grammars,” *Workshop on Dynamical Vision at ICCV*, Beijing, China, 2005.

[14] J. Park, S. Park, and J. K. Aggarwal, “Model-based human motion tracking and behavior recognition using hierarchical finite state automata,” *Proceedings of International Conference on computational Science and Its Applications*, Assisi, Italy, pp. 311- 320, 2004.

[15] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, Vol. 77, No. 2, 1989.

[16] D. Ron, Y. Singer, and N. Tishby, “The power of amnesia,” *Advances in Neural Information Processing Systems*, Morgan Kauffmann, New York, pp. 176- 183, 1994.

[17] R. Rosenfeld, “Two decades of statistical language modeling: where do we go from here,” *Proceedings of the IEEE*, Vol. 88, No. 8, pp. 1270- 1278, 2000.

[18] R. Sharma, V. I. Pavlović, and T. S. Huang, “Toward multimodal human-computer interface,” *Proceedings of the IEEE*, Vol. 86, No. 5, pp. 853- 869, 1998.

[19] A. Vinciarelli, S. Bengio, and H. Bunke, “Offline recognition of unconstrained handwritten texts using HMMs and statistical language models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 6, pp. 709- 720, 2004.

[20] L. Wang, w. Hu, and T. Tan, “Recent developments in human motion analysis,” *Pattern Recognition*, Vol. 36, No. 3, pp. 585- 601, 2003.

[21] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: real-time tracking of the human body,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 780- 785, 1997.

[22] J. Yamato, J. Ohya, and K. Ishii, “Recognizing human action in time-sequential images using hidden Markov model,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 379- 385, 1992.

[23] J. Yang, Y. Xu, and C. S. Chen, “Human action learning via hidden Markov model,” *IEEE Transactions on System, Man, and Cybernetics*, Vol. 27, No. 1, pp. 34- 44, 1997.