Gesture Spotting and Recognition for Human–Robot Interaction

Hee-Deok Yang, A-Yeon Park, and Seong-Whan Lee, *Senior Member, IEEE*

***Abstract*—Visual interpretation of gestures can be useful in accomplishing natural human–robot interaction (HRI). Previous HRI research focused on issues such as hand gestures, sign language, and command gesture recognition. Automatic recognition of whole-body gestures is required in order for HRI to operate naturally. This presents a challenging problem, because describing and modeling meaningful gesture patterns from whole-body gestures is a complex task. This paper presents a new method for recognition of whole-body key gestures in HRI. A human subject is first described by a set of features, encoding the angular relationship between a dozen body parts in 3-D. A feature vector is then mapped to a codeword of hidden Markov models. In order to spot key gestures accurately, a sophisticated method of designing a transition gesture model is proposed. To reduce the states of the transition gesture model, model reduction which merges similar states based on data-dependent statistics and relative entropy is used. The experimental results demonstrate that the proposed method can be efficient and effective in HRI, for automatic recognition of whole-body key gestures from motion sequences.**

***Index Terms*—Gesture spotting, hidden Markov model (HMM), human–robot interaction (HRI), mobile robot, transition gesture model, whole-body gesture recognition.**

I. Introduction

ROBOTICS research is currently supported in a dynamic environment. Traditional robots were used in factories for the purpose of manufacturing, transportation, and so on. Recently, a new generation of "service robots" has begun to emerge [31].

The United Nations (UN), in their recent robotics survey, divided robotics into three main categories: industrial, professional service, and personal service robotics [27]. Industrial robotics is most commonly deployed. The professional service and personal service robots assist people in the pursuit of their goals [7], [25].

Human motion sequences are typically analyzed by segmenting them into shorter motion sequences, called gestures [9]. Gestures are most commonly used for communication among humans, reducing the chances of misclassifying static poses, by using continuous information. Gestures can be divided into two gestures, a communicative gesture (a key gesture or a meaningful gesture) and a noncommunicative gesture (a garbage gesture or a transition gesture) [9]. A key gesture is motion that carries an explicit meaning to express goals, and a transition gesture is motion that connects key gestures to cater to subconscious goals. Gestures can be categorized according to their functionality, as shown in Table I [20].

TABLE I
GESTURE TAXONOMY DEFINED BY RIME AND CHIARATURA [20]

Types	Meaning	Examples
Symbolic Gesture	Gestures have a single meaning within each culture.	Sign Language, Command Gesture
Deictic Gesture	Gestures direct the listener's attention to specific events or objects in the environment.	Pointing Gesture
Iconic Gesture	Gestures represent meaningful objects or actions.	Predefined Gesture
Pantomimic Gesture	Gestures that depict objects or actions, with or without accompanying speech.	Mimic Gesture

The problem of recognizing human motion is divided into two components: segmentation and recognition. The gesture segmentation is the task of finding the start and end boundary points of a legitimate gesture. The gesture segmentation is also called gesture spotting. The gesture recognition is the task of matching the segmented gestures against a library of predefined gestures, to determine which class it belongs to. The task of locating meaningful key gestures from a human motion sequence is called key gesture recognition [13]. The difficulty in gesture spotting is that gesture occurrences vary dynamically, in both shape and duration.

Gesture segmentation using continuous video was explicitly attempted by Lee and Kim [13]. They proposed explicit use of a threshold model corresponding to connecting patterns between key gestures. Later, Barbic *et al.* [2] focused only on the segmentation problem, and proposed three methods, based on principal component analysis (PCA), probabilistic PCA, and the Gaussian mixture model (GMM). Although only key gestures are generally of interest, there are as many transition gestures in human motion.

Starner *et al.* [21] used the hidden Markov model (HMM) for American Sign Language recognition among a variety of modeling tools. HMM is well known for its capability in modeling spatio-temporal variability [17]. In this method, HMMs are trained to model the variability of key patterns. They did not create a model for transition gestures occurring between

Manuscript received March 16, 2006; revised September 10, 2006. This paper was recommended for publication by Associate Editor H. Zhuang and Editor K. Lynch upon evaluation of the reviewers' comments. This work was supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea. This paper was presented in part at the 7th IEEE International Conference on Automatic Face and Gesture Recognition, Southampton, U.K., April 2006.

H.-D. Yang and S.-W. Lee are with the Department of Computer Science and Engineering, Korea University, Seoul 136-713, Korea (e-mail: hdyang@image.korea.ac.kr; swlee@image.korea.ac.kr).

A.-Y. Park was with the Department of Computer Science and Engineering, Korea University, Seoul 136-713, Korea. She is now with LS Industrial Systems Corporation, Seoul 100-753, Korea

Color versions of Figs. 1–7, 10, and 12–22 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2006.889491

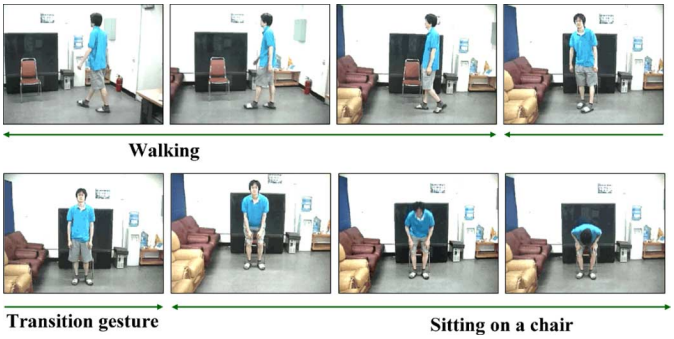

Fig. 1. Motion example consisting of a sequence of key gestures and transition gestures.

key gestures. However, to date, few previous researches put emphasis on explicit modeling of transition gestures. A good gesture recognizer attempts to process this transition motion in a systematic manner. The goal of this paper is to model transition gestures explicitly. Fig. 1 shows a sample video sequence containing several atomic gestures.

Gesture recognition for the mobile robot imposes several requirements on the system. First of all, the gesture-recognition setup is required to be fast enough to fit in the mobile robot environment, as both the human and robot may be moving while a gesture is performed. The system may not assume a static background or a fixed location of the human performing a gesture [23].

Fig. 2 shows a block diagram of the proposed gesture-spotting and recognition method. The first two stages of feature extraction and feature clustering constitute the important preprocessing of feature processing stage. The output is a sequence of feature vectors. This sequence is analyzed next in the spotting and recognition module. For this task, a set of HMMs is required.

The remainder of this paper is organized as follows. Section II reviews related work, and is divided into three categories. Section III describes human motion and feature vector representation. Section IV explains key techniques of the pattern model, including the design of gesture models, a transition gesture model, and a spotter network, followed by the computational algorithm. Section V presents a set of test results and discusses their implications. Section VI concludes this paper.

II. Related Work

There are many gesture-recognition systems for HRI. However, automatic recognition of gestures from a whole-body motion sequence for HRI is rare. Major approaches relating to gesture recognition can be divided into two categories: template matching-based [3], [12], [14], [17], [18], [24], [30] and state-space-based approaches [11], [13], [15], [19], [23]. This section reviews previous work reported in the literature. For a comprehensive review of motion analysis, in a wider perspective, reference can be made to the review paper by Aggarwal *et al.* [1]. For an HRI framework, reference can be made to the review paper by Fong *et al.* [7] and by Thrun [25]. For human–computer interaction, reference can be made to the review paper by Pavlovic *et al.* [16].

A. Template Matching-Based Approaches

Template matching-based approaches assume that gesture models follow a particular pattern that can be modeled as a spatio-temporal template. Generally, the template matching in its pure form cannot be easily applied to the domain of temporal variability, because it is based on the spatial distance between template and input data. This approach is useful when the training set is small and the variance is not great.

Waldherr *et al.* [30] introduced a hand command gesture interface for the control of a mobile robot equipped with a manipulator. A camera was used to track a person and recognize hand gestures involving arm motion. To track a person, the adaptive tracking algorithm was used. The results were compared with two methods; a template and a neural-network (NN)-based method. Four command gestures were experimented with, resulting in a 97% recognition rate. The developed algorithm is integrated in the mobile robot AMELA, which is equipped with a color camera mounted on a pan-tilt unit.

Kortenkamp *et al.* [12] developed a system using a stereo vision system in order to recognize gestures. The system is capable of recognizing up to six distinct gestures, such as pointing and hand signals. The system then interprets these gestures within the context of intelligent agent architectures. Gestures are recognized by modeling the person's head, shoulders, elbows, and hands as a set of proximity spaces. Each proximity space is a small region in the scene for measuring stereo disparity and motion. The robot recognizes gestures by examining the angles between links that connect the proximity spaces. The PRISM-3 system is equipped with a stereo camera, mounted on a pan-tilt head.

Bobick and Davis [3] proposed motion energy image (MEI) and motion history image (MHI) as a template for recognizing human motion. Apart from the inaccuracy of MHI and MEI, the reliance on templates meant the system suffered from the viewpoint problem.

Earlier than this, Takahashi *et al.* [24] had proposed a spotting algorithm called continuous dynamic programming (CDP), in order to recognize seven different body gestures. In this method, a set of standard sequence patterns corresponding to key gestures were represented in the form of a spatio-temporal vector field, and compared with an input sequence using the CDP matching algorithm.

Pineau *et al.* [18] developed a mobile robotic assistant to assist elderly individuals with mild cognitive and physical impairments, as well as support nurses in their daily activities. Their robot communicates with elderly individuals using speech recognition and touch-screen. The robot can detect and track the person and predict the motion of a human.

Perzanowski *et al.* [17] developed a multimodal HRI on the mobile robot platform. In their research, deictic gestures can be recognized using a graphical user interface and pointing device on a personal digital assistant (PDA) or some other form of end-user terminal (EUT), such as a touch-screen. They showed how various modes of their interface can be used to facilitate

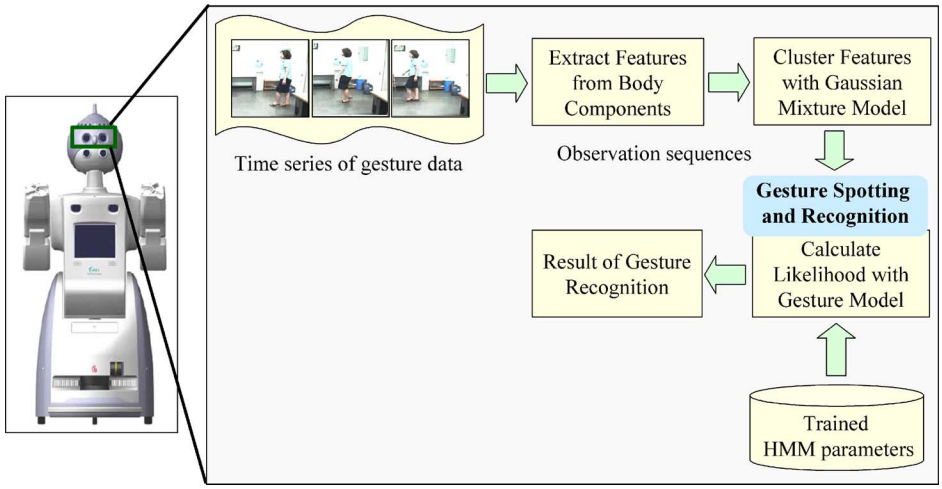

Fig. 2. Block diagram of the proposed gesture-spotting system.

communication and collaboration using a multimodal interface. They overcame the ambiguity of gestures using speech and other interfaces. Due to the use of PDA or EUT, however, a complex architecture was needed for their application.

Nakauchi *et al.* [14] developed a robot which can stand in line with other people. It is one of the most highly socialized and crucial skills required for robots which execute tasks in peopled environments. They implemented the proposed method on Xavier Robot. Xavier has a front-pointing laser light striper with 30° fields of view and two monochrome cameras on a directed perception pan-tilt head. They modeled humans using personal space, which is a person's own territory. In order to stand in line with other people, they detect humans using disparity information and they calculate the direction of bodies without recognizing gestures.

B. State-Space-Based Approaches

State-space-based approaches are used to model systems whose internal states change over time, based on a string of input symbols [9]. These states are connected with each other by certain probabilities. Any sequence as a composition of these symbols is considered a tour through various states. In state-space-based approaches, time variance of a symbol is no longer a problem, because each state is able to visit itself [9].

1) NN Approach: As large data sets become increasingly available, more emphasis is placed on NNs. Two approaches of representing temporal information exist. The first is to use a recurrent NN, and the second is to use a multilayer feedforward network, with a sophisticated preprocessing architecture.

Stiefelhagen *et al.* [23] used the 3-D position of head and hands in order to recognize gestures. Skin color was used to detect and track head and hand regions. A stereo camera was used to capture the data. Two NNs were constructed, one for pan and another for the tilted angle of a head pose. These NNs process the head's intensity and disparity. The combined use of depth and gray information proved better result than either gray or depth information. The algorithm was integrated into the ARMAR, a humanoid robot with two arms and 23 degrees of freedom.

2) HMM Approaches: The HMM is one of the most successful and widely used tools for modeling signals with spatio-temporal variability [19]. This tool has been successfully applied to speech recognition, and has been extended to other applications such as gesture recognition, protein modeling, and so on.

Nickel and Stiefelhagen [15] used HMM for recognizing the 3-D pointing gestures for HRI using a stereo camera. When estimating the performance of the pointing direction, two approaches were compared: the head-hand line and the 3-D forearm direction. In the research, the head-hand line was a better feature in recognizing a 3-D pointing gesture.

Lee and Kim's method [13] is considered to be the first to consider transition gestures as a pattern of separate modeling. The method was constrained to analyzing the 2-D trajectory of a hand, without considering the number of samples when merging two states; this is considered to be a considerable weakness of the method.

More recently, Kahol *et al.* [10], [11] attempted segmentation of a complex human motion (e.g., dancing) sequence. An algorithm called hierarchical activity segmentation (HAS) was proposed. This algorithm employed a dynamic hierarchical layered structure for representing the human anatomy, using low-level motion parameters in order to characterize simple motions bottom-up. Their method consists of two steps. Potential gesture boundaries is first recognized with three cues, then these potential gesture boundaries are fed to the naive Bayesian classifier to find the correct gesture boundary. The coupled HMM (cHMM) was used for individual gesture patterns to spot dance sequences. Their method used 3-D information as features.

In many other researches, only key gestures are generally of interest. However, there are many transition gestures in human motion. In order to spot key gestures exactly, we model transition gestures explicitly.

III. Gesture Feature Representation

Given a set of video frames, it is required to detect and track a human subject over time. As in many pattern-recognition problems, a good representation of the target object is a key to success. This section concerns the technique of representing body features.

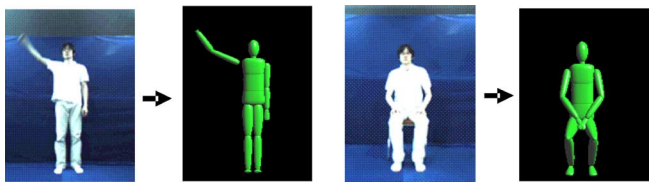

Fig. 3. 3-D whole-body modeling by detecting body components in image sequence.

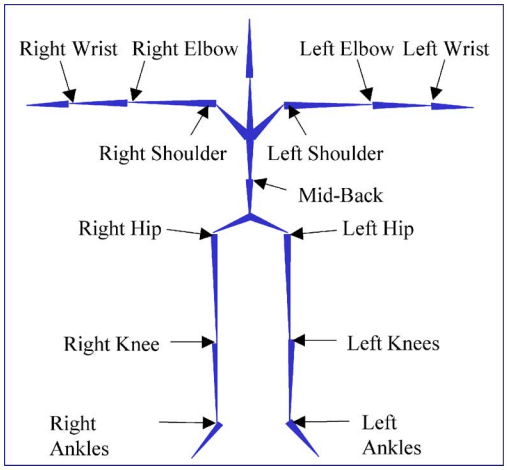

Fig. 4. The 13 body components.

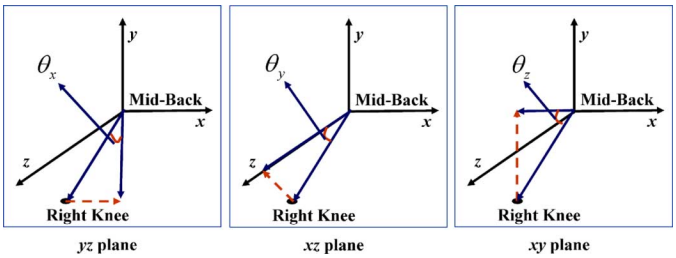

Fig. 5. Thirteen feature points extracted from each body component and the definition of angle features.

A. Feature Extraction

The first step of gesture video processing is to detect a human subject in each of the frame sequences. Then the 3-D pose of the human body is estimated using the pose-reconstruction method described in [32]. It is based on the 3-D model as shown in Fig. 3. Given an image frame, we identify individual body components. Fig. 3 shows two samples of 3-D component detection results in terms of 3-D models simulating the subject's gesture.

The information about body components in 3-D allows us to locate various structural feature points around the body. Thirteen feature points are approximated, as shown in Fig. 4 [1].

Representing the feature points can be done in many ways, namely, the spatial location and velocity. However, these are sensitive to translation and rotation, respectively. Instead, we measured the angle from the vertical axis which measured at the center of the mid-back to each of the feature points. We project the coordinates of the each body components into x, y, and z axes, respectively, to extract the features (see Fig. 5).

Then we can write the feature vector corresponding to the frame at time t as follows:

$$X_t=[F_{L\_\text{shoulder}}, F_{L\_\text{elbow}}, F_{L\_\text{wrist}}, \ldots, F_{R\_\text{knee}}, F_{R\_\text{ankle}}]^T,$$
$$X_t \in \Re^{36}, \quad t>0$$
$$F_k=[\theta_x, \theta_y, \theta_z]$$

where F_k is the three angle values of the 3-D human body component at k.

Once the feature vector is defined, we can define a gesture as an ordered sequence of feature vectors $X = \mathbf{x}_1\mathbf{x}_2\mathbf{x}_3\cdots\mathbf{x}_T$. These feature vectors will then be clustered in the next step.

B. Feature Clustering

Human motion, including gestures, can be represented as a sequence of feature vectors. The sequence of feature vectors constitutes a complex spatio-temporal trajectory in multidimensional space. The motion trajectory is considered as a sequence of vectors combining meaningful key gestures and meaningless transition gestures.

Let us write $\mathbf{x}_t \in \Re^n$ as a feature vector in the n-dimensional feature space $\Re^n$. Then, a whole trajectory can be written as a sequence of feature vectors as $X = \mathbf{x}_1\mathbf{x}_2\mathbf{x}_3\cdots\mathbf{x}_T$. Fig. 6 shows sample trajectories of two gestures in low 3-D subspace; PCA was done for visualization.

The first step of feature processing to the gesture analysis is partitioning the feature space. To achieve this goal, we divide a set of feature vectors into a set of clusters. This allows us to model the trajectory in the feature space by one in the cluster space. Different gestures have different cluster trajectories, even though many clusters are shared by other gestures.

As a means of clustering feature vectors, we employed the technique of expectation-maximization (EM)-based GMM [5]. This algorithm leads us to a space partition $C = \{C_1, \ldots, C_k\}$, where k is the number of clusters, and each cluster corresponds to a region in the feature space. In this method, a feature vector x can be modeled by a GMM-based distribution function as

$$P(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)P(C_k)}{p(\mathbf{x})} \tag{1}$$

where $p(\mathbf{x}|C_k)$ is the probability density function (PDF) of class C_k, evaluated at x, $P(C_k)$ is the prior probability for class C_k, and $p(x)$ is the overall PDF evaluated at x. Note that $p(\mathbf{x}|C_k)$ is modeled by a multivariate Gaussian

$$p(\mathbf{x}|C_k)=\frac{1}{(2\pi)^{n/2}\,|V_k|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\mu_k)^t V_k^{-1}(\mathbf{x}-\mu_k)\right] \tag{2}$$

where μ_k is the mean and V_k is the covariance matrix of the cluster C_k..

We compute the cluster index of given feature vector, then the calculated cluster index is used as input as the observation symbol in the HMM. We have to specify the number of clusters C for each execution of the GMM, and we usually do not know the best number of clusters in a data set. We chose $C = 25$ based on the amount of data we have. Each gesture has a distinct sequence of cluster indices. A general observation is that different gestures have different trajectories in the cluster space, while

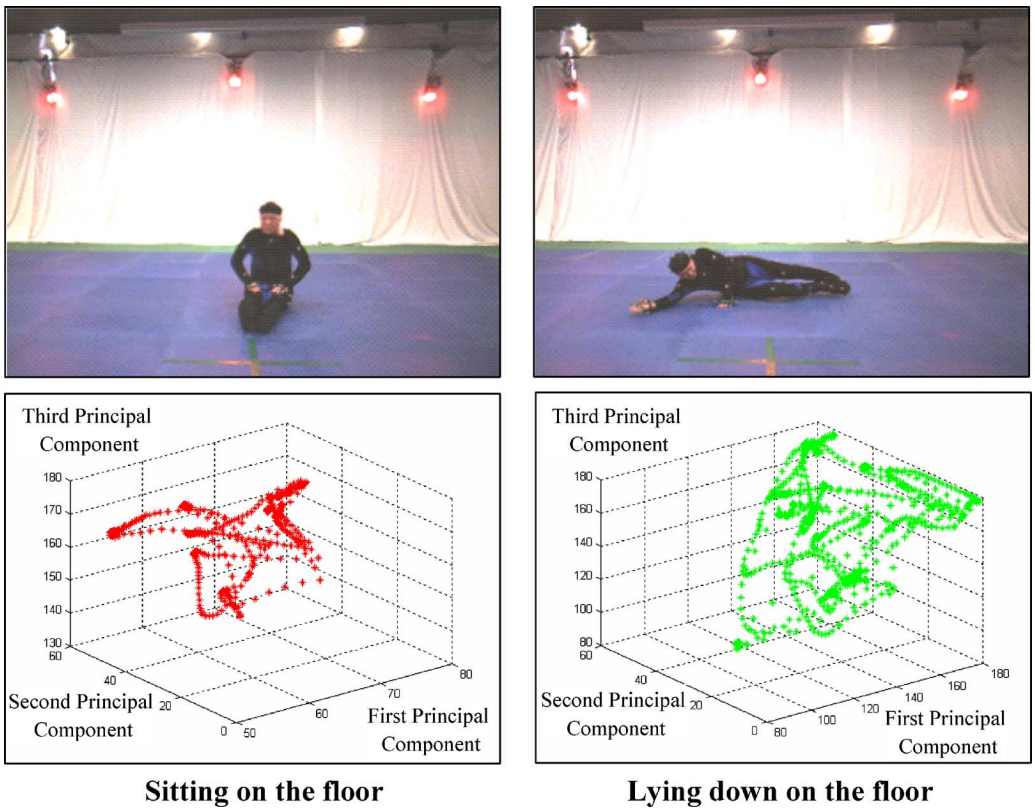

Fig. 6. Feature trajectories of two gestures in a reduced-dimensional subspace.

the same gestures show very similar trajectories (see Fig. 7). Note that several occurrences of walking follow the same sequence of clusters except for the temporal variations: start and end points and durations in a cluster. There are no such common paths among different gestures. Even though different gestures may have the same cluster indices at a particular time, each gesture shows a highly distinct cluster sequence.

IV. Key Gesture Spotting

The task of key pattern spotting is to find the start and end boundary points of a legitimate gesture while ignoring the rest. It can be regarded as a simplified task of small vocabulary recognition. The two key issues in spotting are how to model of key patterns discriminately and how to model transition (nonkey) patterns effectively. Here, we discuss the solutions of key gesture spotting in detail.

Key patterns are modeled by gesture HMMs, and all transition patterns are modeled by a single sophisticated HMM. Unlike the key patterns, however, it is not easy to obtain a training set of transition patterns because there are infinite varieties of transition motions. The primary focus of this section is this issue and the method of actually locating key gestures.

A. Gesture Model

A gesture is part of a spatio-temporal trajectory. It has a stereotypical trajectory that is consistent over a wide range of variations. Such a pattern can be modeled effectively by an HMM. When applied to key gesture pattern modeling,

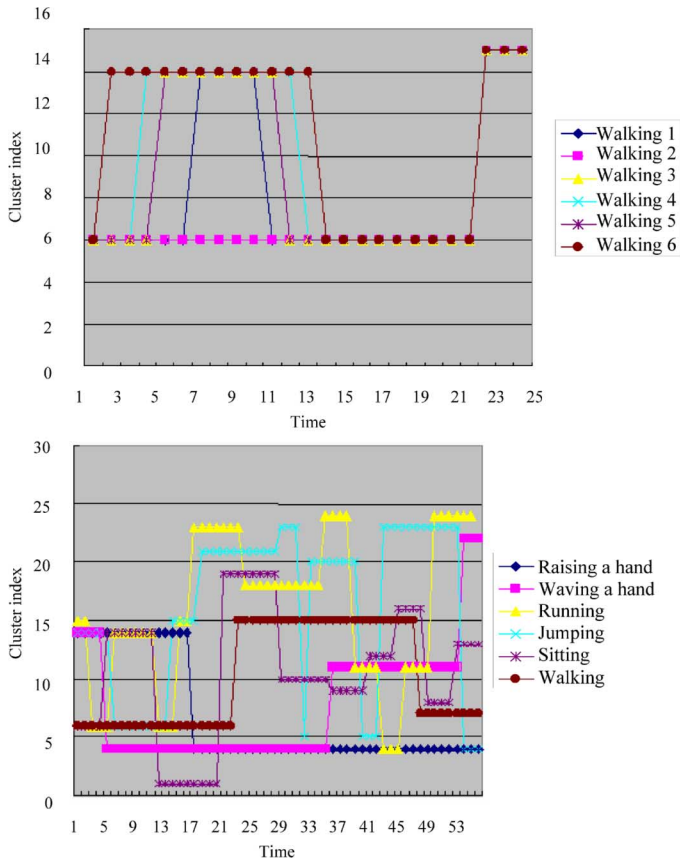

Fig. 7. Cluster trajectories of the same gestures (top) and different gestures (bottom).

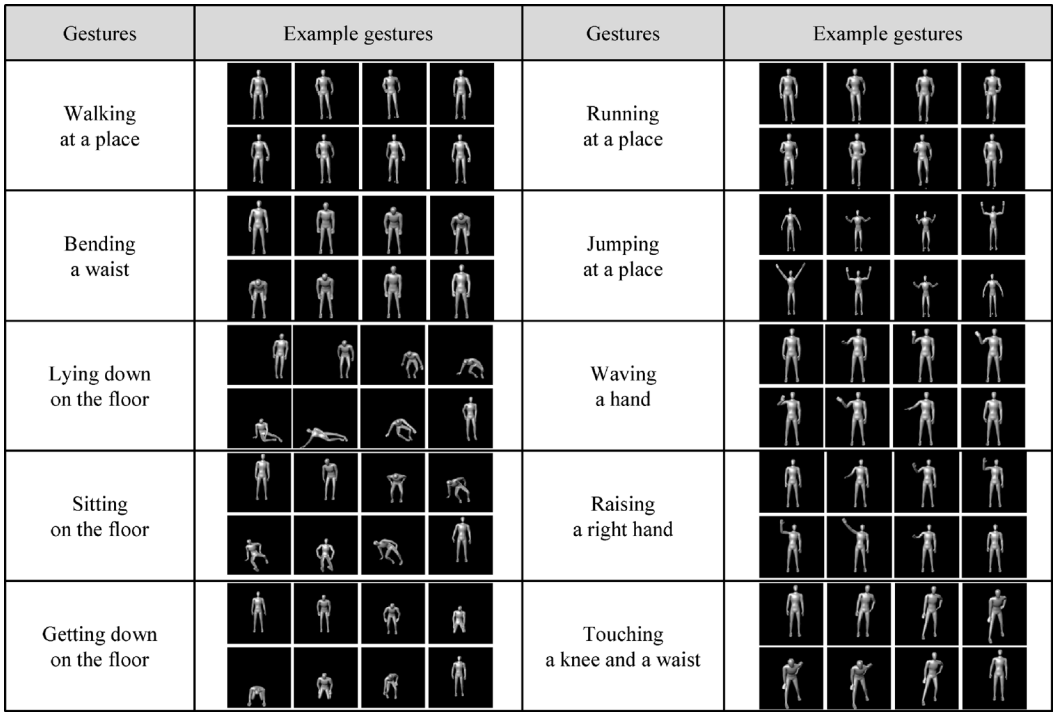

Fig. 8. Examples of ten gestures.

each state of an HMM represents the local segmental characteristics of the gesture, while the state transitions represent the sequential order structure of the segments in a gesture's trajectory. Since there is a strong temporal constraint in the gesture patterns, the most natural choice is left–right HMM with a strict left-to-right transition constraint, order structure, and no backward transition.

Gesture HMMs are trained using the Baum–Welch algorithm [19]. For this, we have prepared a set of isolated gesture data. Since the set is insufficient, we have augmented the set with gesture variations derived from eigengestures and Gaussian random noise [28]. An eigengesture is an eigenvector derived from the covariance matrix of the probability distribution in the vector space of human gestures. The number of HMM states depends on the average gesture signal length, complexity, and the variability of the pattern. Various algorithms are considered and presented to find the number of states of HMM. In this research, minimum description length (MDL) based on the state-splitting method is used [22]. The formal procedure for searching the number of states goes as follows.

Step 1) Start with an HMM with the number of state, equal to 1.

Step 2) Train the model on the training set using Baum–Welch algorithm.

Step 3) Select the split which gives the maximum increases in the likelihood on a constrained subset of parameters.
The increase in the likelihood $G(i), i \in S$ for a split is can be given by the equation shown at the bottom of the page, where $\tilde{S}$ is the set of the two states $(\tilde{i}, \tilde{i}_2)$ resulting from splitting state i.

Step 4) Determine the likelihood increase for the complete model by training a model after splitting the state with the Baum–Welch algorithm.

Step 5) If the increased likelihood is larger than the MDL penalty difference, then split and go to step 2. Stop otherwise.

There are ten different gestures considered in this study. They are shown in Fig. 8.

B. Transition Gesture Model

Unlike the stereotypic gesture models, there is no constraint on the remaining transition gesture patterns. A transition gesture is any motion trajectory or any part of it other than

$$G(i) = -\sum_{t=1}^{T}[p(s_t = i, s_{t-1} = i|O,\theta)\log p(s_t = i, |O,\theta)\log p(s_t = i|s_{t-1} = i,\theta) + p(s_t = i, |O,\theta)] + \sum_{t=1}^{T}\left[\sum_{k,j\in\tilde{S}} p(s_t = j, s_{t-1} = k|O,\theta)\log p(s_t = j|s_{t-1} = k,\theta) + \sum_{j\in\tilde{S}} p(s_t = j|O,\theta)\log p(s_t = j|O,\theta)\right]$$

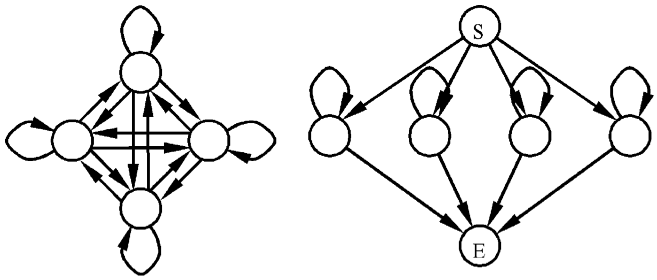

Fig. 9. Two types of ergodic model structure. (a) Ergodic or fully connected topology. (b) Simplified ergodic structure with two dummy or null states and fewer transitions.

gestures. Therefore, we choose to design a type of ergodic or fully connected model in which each state of the model can be reached from all other states. Fig. 9(a) illustrates the ergodic model topology. As the number of states increases, however, the number of edges grows by its second order and soon becomes unmanageable in our case. Thus a well-known method is to use the topology of Fig. 9(b). Here, two dummy states are introduced, enabling a much simpler structure. A dummy state is also called a null state which produces no observation symbols [5].

The formal procedure for constructing the transition gesture model goes as follows.

Step 1) Duplicate all states j from all gesture HMMs, each with an output distribution $b_j(k)$ where k is the output symbol. Let S_{G} be the set of all states from all gesture models. These states are highly adapted to the local gesture patterns. Before using these, we smooth the output distributions using a Gaussian filter and then apply the floor smoothing.

Step 2) Attach the original self-transition to each state with the same probability.

Step 3) All duplicate states have only one outgoing transition reaching the dummy state ET in Fig. 10. Their transition probabilities are simply given by

$$\begin{aligned} a_{jE} &= 1 - a_{jj}, \quad \text{for all } j \in S_{\mathrm{G}} \\ a_{Sj} &= \frac{1}{|S_{\mathrm{G}}|}, \quad \text{for all } j \in S_{\mathrm{G}} \end{aligned} \tag{3}$$

where a_{jE} is the transition probability from state j to the dummy end state, a_{Sj} is the probability from the dummy start state to state j, and $|S_{\mathrm{G}}|$ is the number of states in S_{G}.

The two dummy states observe no symbol and are passed without time delay. They just serve as a branching point and a merging point, respectively. Therefore, every state can be reached from every state in a single transition. Thus, the transition gesture model is an ergodic model. Due to the smoothed output parameters, the model can be used to model arbitrary subpatterns appearing in any order (see Fig. 10).

Since the gesture models are optimized for the target patterns, their likelihood will be greater than the transition gesture, as well as other nontarget models for the target patterns. Conversely, if the gesture models are specialized to target patterns,

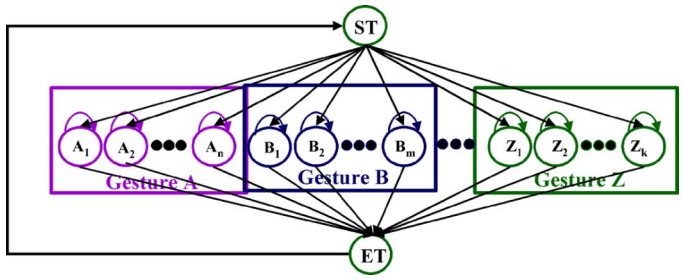

Fig. 10. Transition gesture model.

their likelihood to transition gesture pattern will drop below that of the transition gesture model, which allows any arbitrary patterns thanks to the smoothed distributions. The transition gesture model represents every possible pattern. Thus, if there is a segmental region (or a subsequence X) in an input sequence and a certain model's likelihood is greatest or

$$P(X|\lambda_{\mathrm{gesture}(k)}) > P(X|\lambda_{\mathrm{transition\ gesture}}) \tag{4}$$

then we can safely assume that X is the kth gesture. In this sense, the transition gesture model provides a confidence measure as a function of input data, that is, an adaptive threshold for gesture spotting.

C. Model Reduction

The number of states in the transition gesture model is equal to the number of states in all gesture models combined, except the null start and null final states. This means that the transition gesture model size increases in proportion to the number and size of the gesture models. Since there are many states with similar output distributions, an increase in the number of states is nothing but a waste of time and space. There is a strong need for reducing the states.

Among others, relative entropy is a useful measure of distance between two distributions in HMM states. For two distributions $P = \{P(i)|i \in V\}$ and $Q = \{Q(i)|i \in V\}$ where V is the set of output symbols, the symmetric relative entropy $D(P\|Q)$ is written as

$$D(P\|Q) = \frac{1}{2}\sum_i \left(P(i)\log\frac{P(i)}{Q(i)} + Q(i)\log\frac{Q(i)}{P(i)} \right). \tag{5}$$

The proposed state-reduction procedure is based on (5), and given as follows.

Step 1) Compute the symmetric relative entropy for each pair of distributions P_m and P_n of states m and n, respectively, as

$$\begin{aligned} &D(P_m\|P_n) \\ &= \frac{1}{2}\sum_i \left(P_m(i)\log\frac{P_m(i)}{P_n(i)} + P_n(i)\log\frac{P_n(i)}{P_m(i)} \right), m, n \in S_{\mathrm{G}}. \end{aligned} \tag{6}$$

Step 2) Find the state pair $(\hat{m}, \hat{n})$ with the minimum relative entropy $D(P_{\hat{m}}\|Q_{\hat{n}})$.

Step 3) Merge the two states. This is done by computing an interpolated distribution of $P_{\hat{m}}$ and $P_{\hat{n}}$ instead of the simple average of

$$P_{(m,n)}(i) = \frac{P_m(i) + P_n(i)}{2} \tag{7}$$

we introduce the interpolation weight based on the expected number of symbol observations in state i

$$\alpha = \frac{\sum_j \gamma_t^{g_1}(m)\delta_{v_i o_t^{g_1}}}{\sum_j \gamma_t^{g_1}(m)\delta_{v_i o_t^{g_1}} + \sum_j \gamma_t^{g_2}(n)\delta_{v_i o_t^{g_2}}} \tag{8}$$

where g_1, g_2 are two gesture models from which the two states m and n came, $\sum_t \gamma_t^g(m)$ is the expected number of observations in state m in g, δ_{ab} is

$$\text{Kronecker delta} = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases}$$

and v_i, o_t^g are the ith symbol and the symbol observed at time t by model g.

Note that the variable $\gamma_t^g(m)$ can be computed using the forward and backward variables of the Baum–Welch procedure. Note that in the case of multiple training data, a new summation over the class samples can be introduced in each of the three summations above. Now the resulting distribution can be obtained by

$$P_{(m,n)}(i) = \alpha P_m(i) + (1 - \alpha)P_n(i). \tag{9}$$

The two old states are replaced by a single state, say $m' \leftarrow (m,n)$, and the statistics for the expected number of observation are added to produce

$$\Gamma_{m'i} = \sum_t \gamma_t^{g_1}(m)\delta_{v_i o_t^{g_1}} + \sum_t \gamma_t^{g_2}(n)\delta_{v_i o_t^{g_2}} \tag{10}$$

which will be used in the future merge with another state.

Step 4) If the number of states is greater than a threshold value, then go to step 1. Stop otherwise.

The procedure of model reduction is presented graphically in Fig. 11.

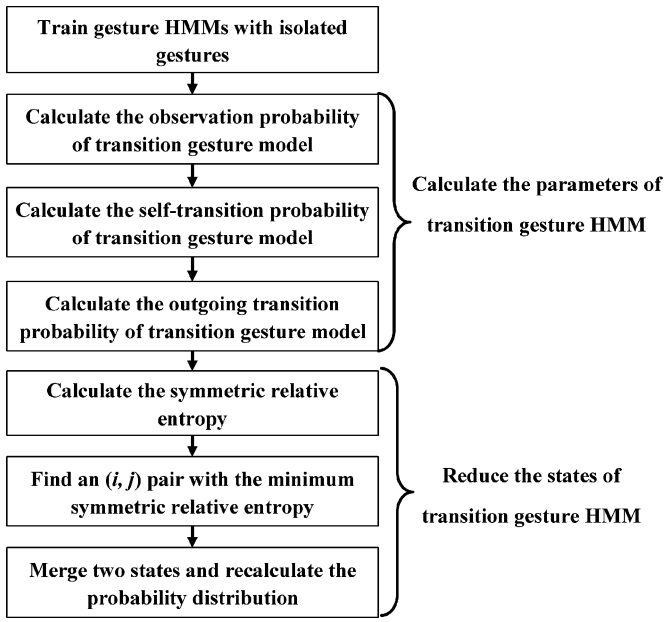

Fig. 11. Procedure of the model reduction.

D. Key Gesture Spotting Model

In continuous human motion, gestures appear intermittently with transition connecting motion. There is no specific order among different gestures and any knowing when any gesture starts to appear and ends. We have defined the meaningless intergesture pattern as the transition gesture. Then one way to define the alternating sequence of gestures and transition gesture is to construct a cascade connection of gesture HMMs and a transition gesture HMM repeatedly. A more effective structure is a circular interconnection of HMMs: key gesture HMMs and then one or more transition gesture HMMs which are then connected to the start of the gesture HMMs. In this research, we designed the network shown in Fig. 12. As shown in Fig. 12, we can easily expand the vocabulary by adding a new key gesture HMM model and rebuilding a transition gesture model.

Our design employed two copies of the transition gesture HMM; one is for the initial motion before the first gesture, and the other is for the rest transition gesture motions between gestures. This allowed the system to detect the first gesture better in a series of tests. The ten gesture models shown in Fig. 12 are illustrated as used in the current implementation; a strictly left–right model with a varied number of states.

The transition gesture model is constructed using the states of all gesture HMMs. Since they were merged successively to result in a small number of states, they are poor models for all gestures. But they are still all-gesture models. This implies that the model gives a lower threshold to the model likelihood for a motion segment to be accepted as a gesture. Let Λ be a set of gesture HMMs. Then, if

$$\exists g : p(X|\lambda_g) > p(X|\lambda_{\mathrm{G}}), \quad g \in \Lambda$$

then X can possibly be as gesture g. Furthermore, if g happens to be the most likely, then X is estimated to be gesture g. On the other hand, if

$$\forall g : p(X|\lambda_g) < p(X|\lambda_{\mathrm{G}}), \quad g \in \Lambda$$

then we can say that X cannot be a gesture. Note that the computed likelihood of the transition gesture model can be used as an adaptive confidence measure for spotting a small set of gestures, refer to Fig. 13. With the gesture spotter network, all gestures in human motion can be recognized, and the beginning and the end points are obtained simultaneously.

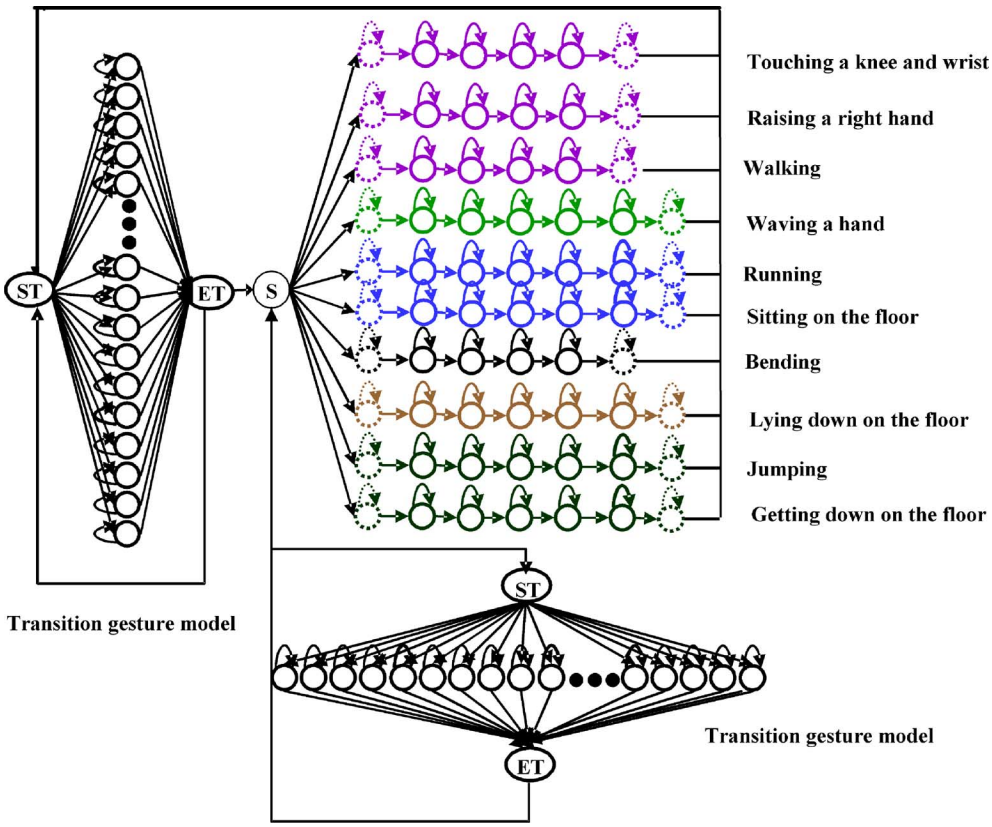

Fig. 12. Key gesture spotting model.

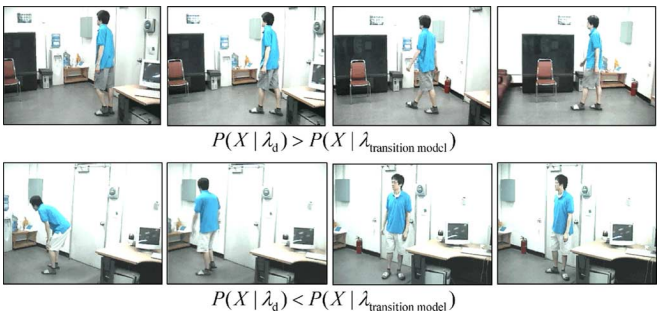

Fig. 13. Likelihood of the transition gesture model is less than the key gesture models (λ_c) given a gesture motion, but greater given a segment of transition gesture movement.

The computation algorithm for the spotting gestures in a given sequence of observations is based on the dynamic programming-based Viterbi algorithm [29]. The spotter model is a network of HMMs, each of which in turn is a small network. By viewing it as a two-level network. we perform Viterbi algorithm at both levels: within individual HMMs and between those HMMs.

The first level is the ordinary Viterbi algorithm and is called the state dynamic programming (DP) as the primary computation is done at the level of states. Given an observation sequence $O_{t_1,t_2} = o_{t_1} o_{t_1+1} o_{t_1+2} \cdots o_{t_2}$, the Viterbi likelihood $p(O_{t_1,t_2}, Q^g_{t_1,t_2} | \lambda_g)$ with $Q^g_{t_1,t_2} = q_{t_1} q_{t_1+1} \cdots q_{t_2}$ being the "best" state sequence in each HMM λ_g can be computed as

$$\begin{aligned}
\delta^g_{t_1,t}(j) &= \max_{Q^g_{t_1,t-1}} P(O_{t_1,t}, Q^g_{t_1,t-1}, q_t = j | \lambda_g) \\
&= \max_{Q^g_{t_1,t-1}} P(O_{t_1,t-1}, Q^g_{t_1,t-1}, o_t, q_t = j | \lambda_g) \\
&= \max_{Q^g_{t_1,t-1}} P(O_{t_1,t-1}, Q^g_{t_1,t-1} | \lambda_g) P(o_t, \\
&\quad q_t = j | Q^g_{t_1,t-1}, \lambda_g) \\
&= \max_i \max_{Q^g_{t_1,t-2}} P(O_{t_1,t-1}, Q^g_{t_1,t-2}, q_{t-1} \\
&\quad = i | \lambda_g) P(o_t, q_t = j | q_{t-1} = i, \lambda_g) \\
&= \max_i \delta^g_{t_1,t-1}(i) \times a^g_{ij} b^g_j(o_t) \qquad (11)
\end{aligned}$$

where a^g_{ij} and $b^g_j()$ are the parameters of model λ_g. Using the relation, we can compute the joint likelihood of an arbitrary subsequence of observations for any models incrementally and very efficiently. The above formula is computed for all times t, for all models g, and for all states j. The boundary condition for the recurrence relation is as follows:

$$\delta^g_{t_1,t_1}(j) = \Delta_{t_1} \times \pi_j b^g_j(o_{t_1}), \quad \text{for all } j, \quad \text{for all } g, \quad \text{for all } t_1 > 0 \qquad (12)$$

where Δ_{t_1} is an initial base, to be explained later. To take a further step in state-level DP, we modified (12) as

$$\delta^g_t(j) = \max_{t_1} \{\Delta_t \pi_j b^g_j(o_t), \delta^g_{t_1,t-1}(j) a^g_{jj} b^g_j(o_t)\}. \qquad (13)$$

This is the Viterbi likelihood of a partial sequence over all possibilities of starting points (t_1). If the initial state likelihood is written in this way, the left-hand side of (11) is none other than $\delta_t^g(t)$, and we can write the recurrence relation as

$$\delta_t^g(j) = \max_i \delta_{t-1}^g(i) a_{ij}^g b_j^g(o_t). \tag{14}$$

The segmentation information, the beginning and the end points of the gesture or transition gesture, can be obtained by keeping the best state leading to the current state as

$$\psi_t^g(j) = \arg\max_i \delta_{t-1}^g(i) a_{ij}^g \quad \forall j, \quad 2 \le t \le T \tag{15}$$

where T represents the length of the input sequence. In the case when the first term in the right-hand side is chosen, then we let the back pointer simply be -1.

The DP of the second level is the maximization among competing HMMs. It is done at the dummy start and end states. The DP is maximization over all possible models linked to the states

$$\Delta_t(j) = \max_g \delta_t^g(N^g), \quad \text{for all dummy states } j \tag{16}$$

where N^g is the final state of model g, and $\delta_t^g(N^g)$ is the likelihood of the final state of model g. This score is fed to the initial states of individual HMMs connected from the dummy states, as noted in (12) and (13).

Recovering the most likely state sequence or the location of gestures in the input sequence, we need to trace back the Viterbi path by following the chain of back pointers $\psi_t^g(j)$.

V. Experimental Results

The proposed approach has been integrated into a mobile robot, T-Rot (Thinking Robot) [26], and evaluated in a series of experiments with the KU Gesture Database (DB) [8] and real-time data captured with a stereo camera, Videre STH-MDCS2 mounted on T-Rot.

A. Robot Platform

The robot platform used in this research is T-Rot, a personal service robot. T-Rot's aim is supporting elderly people. Another important aim is detecting emergency situations such as sitting on the floor, falling down on the floor, and lying down on the floor. To recognize the emergency situation, whole-body gesture recognition is required.

Elderly people are not expert at operating robots. Therefore, T-Rot is required to be able to interact naturally with elderly people, similar to the way human–human interaction takes place. T-Rot has various interaction methods to provide natural interaction between a robot and its users. The HRI of T-Rot includes a speech recognizer, a face recognizer, a gesture recognizer, a speech synthesizer, a facial expression recognizer, and so on.

Performing various interaction methods, T-Rot has several main boards. T-Rot has a main board for vision components such as gesture, face, and expression recognition. As a result, recognition modules for vision components are performed on the same main board, and do not operate simultaneously.

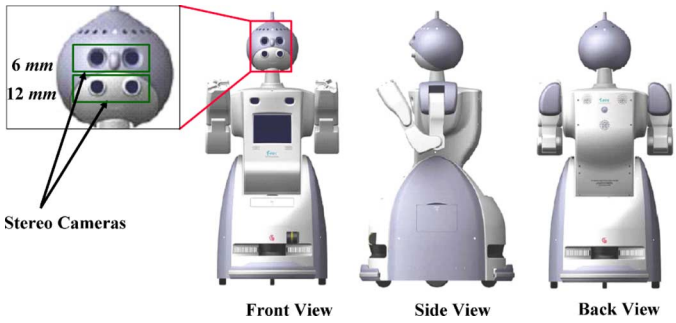

Fig. 14. T-Rot, the robot used in the proposed experiments.

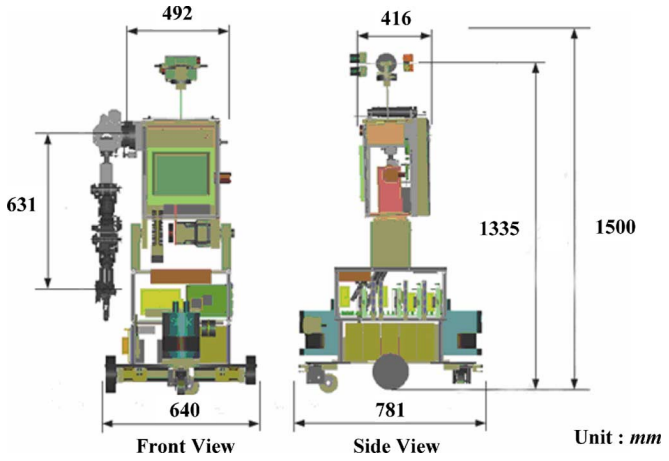

Fig. 15. Mechanical structure and dimensions of T-Rot.

As shown in Figs. 14 and 15, T-Rot is equipped with two stereo cameras, Videre STH-MDCS2, mounted on a pan-tilt unit. The cameras are located on T-Rot's head. The first has a 6 mm focal length, and the second has a 12 mm focal length. The stereo cameras both have a resolution of 320×240. The second camera, with 12 mm focal length, is used to recognize gestures, and the first camera, with 6 mm focal length, is used to recognize a face or object located near T-Rot. In addition, the 2-D laser scanner is attached at the front of T-Rot. This is used for self-localization, navigation, and obstacle detection. Additional safety sensors, such as bumpers and several infrared sensors, are integrated in T-Rot. These sensors are used to detect obstacles.

Fig. 15 shows the dimensions of T-Rot. The height of the lens is approximately 1.3 m from the ground. T-Rot does not move when the gesture recognition module is running, so its body does not tremble. As a result, the captured image from the camera in T-Rot is adequate for recognizing gestures. The optimum distance for recognizing gestures is approximately $2 \sim 3$ m from the subject, as only at this distance can the robot see the subject's whole body (see Figs. 16 and 17).

B. Experimental Data

The performance of the proposed method is measured with the KU Gesture DB [8] which contains 14 full-body gestural

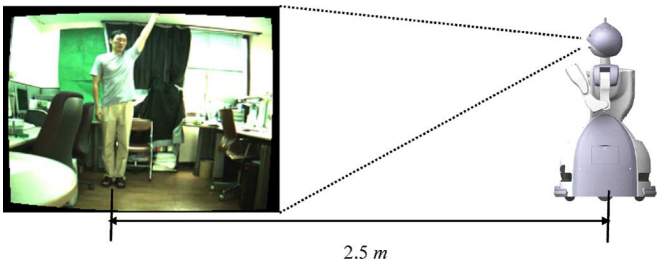

Fig. 16. Experimental environment using real-time data captured with stereo camera, Videre STH-MDCS2 mounted on T-Rot.

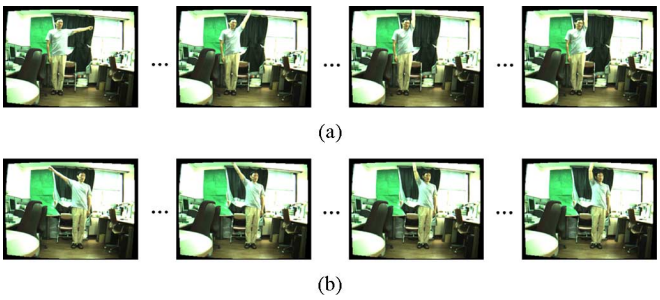

Fig. 17. Examples of raising-a-hand sequence captured with stereo camera, Videre STH-MDCS2 mounted on T-Rot. (a) Raising a left hand. (b) Raising a right hand.

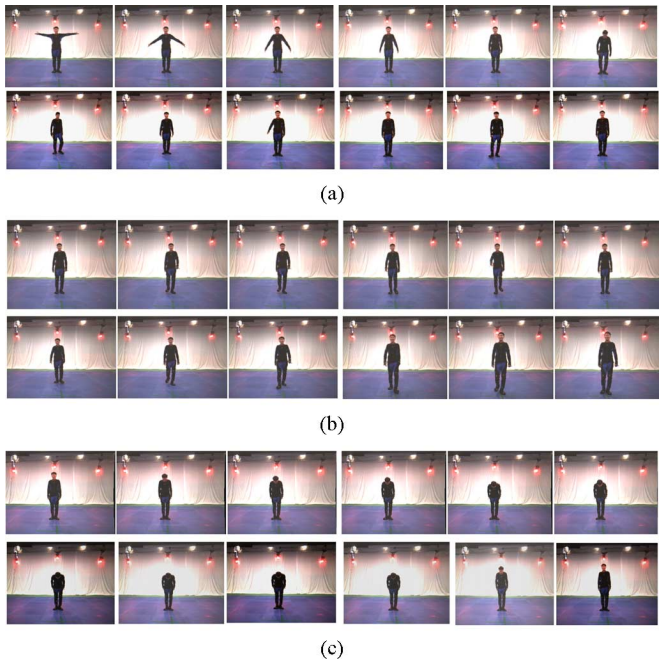

Fig. 18. Sample result of spotting gestures; two gestures have been located after an initial transition motion. (a) Transition gesture sequence from 0 to 9. (b) Walking gesture sequence from 9 to 20 s. (c) Bending gesture sequence from 20 to 25 s.

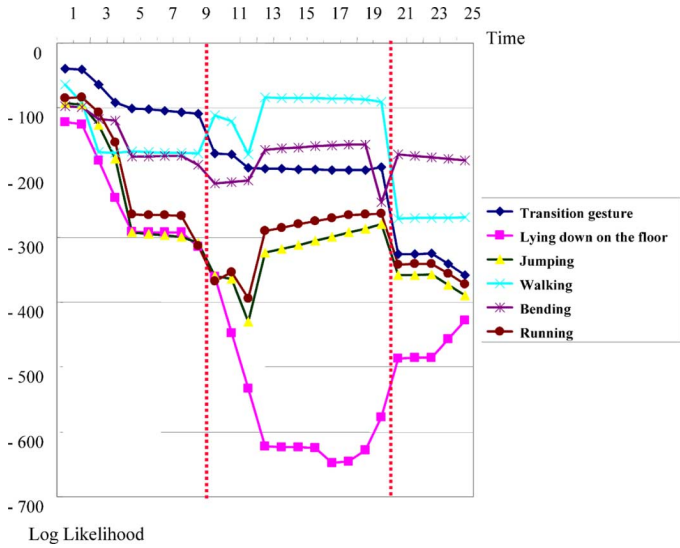

Fig. 19. Temporal evolution of the log-likelihood of the gesture models and a transition gesture model; the vertical broken line marks the end of transition gesture.

motions often encountered in everyday life. Additional experimental data came from the stereo camera, Videre STH-MDCS2, mounted on T-Rot.

The KU Gesture DB was captured at 60 frames per second (fps) from 20 subjects using optical 3-D motion technology. The duration of each gesture is variable from 5 to 10 s. The DB follows the format of hierarchical translation rotation (HTR), one of the major motion file formats. HTR files carry information about the structure of the subjects based on the linked joint model. In this paper, only ten gestures were considered for the experiment.

The real-time data such as walking, sitting on the floor, raising a hand, and bending were captured with a stereo camera, Videre STH-MDCS2, with a resolution set to 320×240. Data was captured for 20 humans of their front and side views. The sequence length of the data was approximately $120 \sim 180$ frames at 30 fps for each sequence. As shown in Fig. 16, the distance for capturing gestures is approximately 2.5 m from the subject.

Fig. 17 shows examples of a raising-a-hand sequence captured with the stereo camera mounted on the mobile robot.

Just like many other pattern-recognition approaches using statistical models, the DB is far from adequate for reliable estimation of HMM parameters [28]. We tried to alleviate the problem by synthesizing gesture variations by adding Gaussian noise to eigengestures.

As shown in Fig. 8, the KU Gesture DB considered only one lateral data such as raising a right hand, waving a right hand, sitting on the floor with right leg first; however, the KU Gesture DB has 3-D information. Therefore, we can generate the symmetric data with graphic tools such as MotionBuilder. In this research, the right lateral gestures are only used to test. However, by adding the left lateral gesture HMMs to the key gesture spotting model described in Section IV-D, we can also recognize the left lateral gestures.

C. Experimental Results With KU Gesture DB

1) A Gesture Spotting Example: Let us first examine the actual result of spotting in a given sample sequence. Fig. 18 shows an example of a motion sequence containing an initial unclassified motion followed by two gestures in succession.

The time evolution of the likelihood values of gesture and transition gesture HMMs is illustrated by the curves of Fig. 19. The transition gesture model has the greatest likelihood during

TABLE II
KEY GESTURE SPOTTING RESULTS. THE FIGURES IN THE FIVE CENTRAL COLUMNS DENOTE SAMPLE COUNTS

Gestures	*N*	N_{Hit}	N_{DE}	N_{SE}	N_{IE}	*R(%)*
Walking	58	55	2	1	2	91.6
Running	62	59	1	2	3	90.7
Bending	54	54	0	0	0	100.0
Jumping	62	61	0	1	1	96.8
Lying down on the floor	61	58	1	2	2	92.0
Waving a hand	60	59	0	1	1	96.7
Sitting on the floor	62	58	2	2	3	89.2
Raising a right hand	62	62	0	0	0	100.0
Getting down on the floor	61	58	1	2	2	92.0
Touching a knee and wrist	60	60	0	0	0	100.0
Total	602	584	7	11	14	94.9

N : Number of input gestures
N_{Hit} : Number of correctly recognized gestures
N_{DE} : Number of deletion errors
N_{SE} : Number of substitution errors
N_{IE} : Number of insertion errors
R : Reliability

the first 9 s. From this, we can infer that there was a transition motion for 9 s which is shown in Fig. 18(a). Then it is followed by a walking gesture. As seen in Fig. 18(b), the subject walked on the floor. It is continuous until the gesture ends at time 20 s, after which, the likelihood precipitates and gives way to another gesture, bending or bowing. The start and the end points of all the gestures and transition gesture were obtained by back-tracking the Viterbi path after the forward pass described in Section IV-D.

2) Spotter Performance: In general, most spotting tasks involve three types of errors, namely, substitution, insertion, and deletion errors. An insertion error occurs when the spotter reports a nonexistent gesture. A deletion error occurs when the spotter fails to detect a gesture existing in the input stream. A substitution error occurs when an input gesture is classified in a wrong category. Following the convention, we measured the system performance in terms of those errors and the reliability. The overall performance is defined as

$$\text{reliability} = \frac{\#\text{ of correctly recognized gestures}}{\#\text{ of input gestures} + \#\text{ of insertion errors}} \times 100\%. \quad (17)$$

Table II shows the detailed result of the spotting test. Note that most of the errors are substitution and insertion errors. The substitution errors imply incorrect classifications, and the insertion errors imply incorrect segmentation and incorrect modeling of gesture patterns. The overall reliability with equal priority is 94.9%, as shown in the bottom row.

For a comparison with known methods, we provide the result by the method proposed by Kahol *et al.* [10], [11]. In Kahol *et al.*'s method, local minima in the segment force were detected as candidate gesture boundaries using three cues. Each of these local minima was then considered as a potential gesture boundary. This sequence of potential gesture boundaries and the gesture boundaries identified by the human were used to train the naive Bayesian classifier. In order to train the naive Bayesian classifier, a human identified the frames containing the gesture boundaries within each sequence. As a result, the potential gesture boundaries generated with three cues [11] and the gesture boundaries identified by the human were used as an input vector to the naive Bayesian classifier. Fig. 20 shows the boundaries detected by Kahol *et al.*'s method for the sequence in Fig. 18. Due to lack of transition gesture modeling, however, this method usually produces many more segmentation points than required.

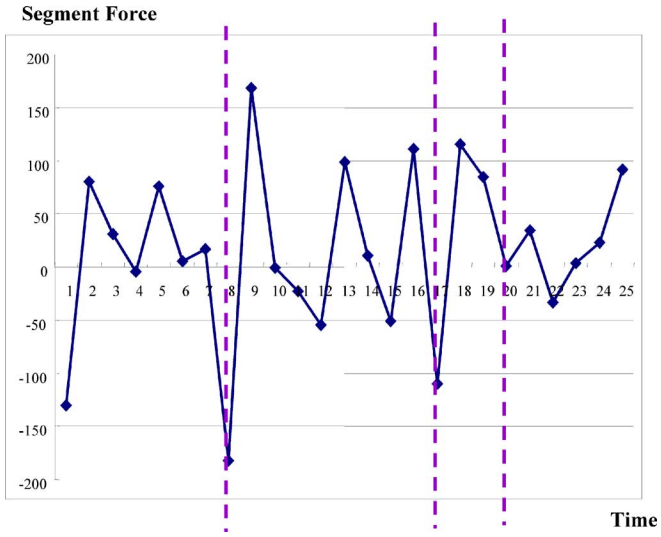

Fig. 20. Gesture boundaries detected by Kahol *et al.*'s method.

TABLE III
KEY GESTURE SPOTTING RESULTS WITH KAHOL ET AL.'S METHOD

Gestures	*N*	N_{Hit}	N_{DE}	N_{SE}	N_{IE}	*R(%)*
Walking	58	55	2	1	5	91.6
Running	62	59	3	0	5	93.6
Bending	54	52	0	2	2	94.5
Jumping	62	59	2	1	3	93.6
Lying down on the floor	61	57	2	2	4	90.4
Waving a hand	60	54	2	4	2	84.3
Sitting on the floor	62	56	3	3	5	83.5
Raising a right hand	62	57	0	5	2	83.6
Getting down on the floor	61	56	3	2	4	87.5
Touching a knee and wrist	60	58	0	2	3	95.0
Total	602	563	17	22	35	91.2

N : Number of input gestures
N_{Hit} : Number of correctly recognized gestures
N_{DE} : Number of deletion errors
N_{SE} : Number of substitution errors
N_{IE} : Number of insertion errors
R : Reliability

The initial transition gesture was recognized. For the remaining two gestures, oversegmentation is evident at 17 s. It is believed that the result is attributed to going without explicit modeling of transition gesture patterns.

Table III shows the test result of Kahol *et al.*'s method. Note that reliability values of some gestures are well below those of the proposed method by more than three percentage points. For the ease of comparison, we provided Fig. 21. This figure alone leads us to believe that the use of the transition gesture model makes a big difference and, without a transition gesture model, the occurrence of insertion errors is unavoidable.

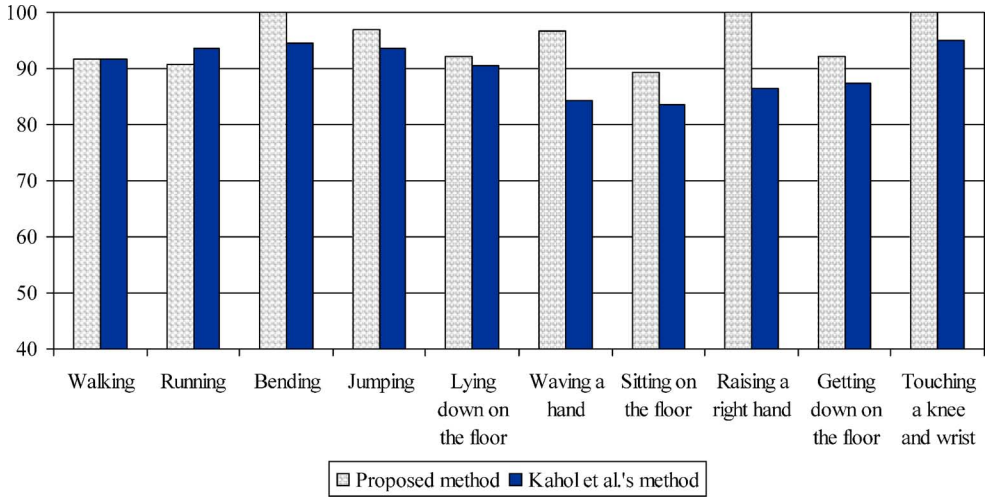

Fig. 21. Comparison chart for the proposed method and Kahol *et al.*'s method.

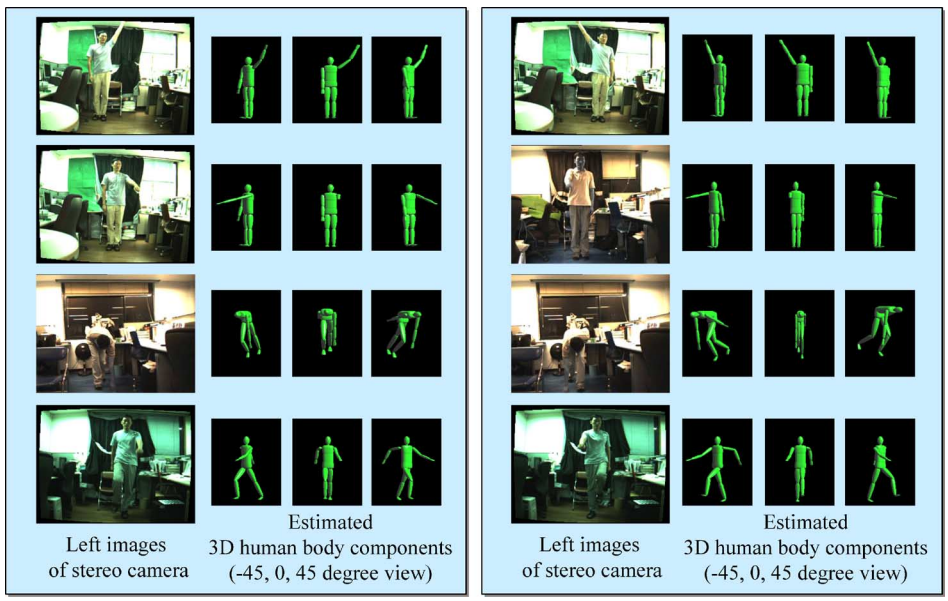

Fig. 22. Examples of estimated 3-D human body component with various real-time data captured with stereo camera, Videre STH-MDCS2 mounted on T-Rot.

3) Isolated Gesture Recognition: For the final set of tests, we divided all the gesture data sets into halves, 50 training samples and 50 test samples. First of all, ten gesture HMMs were trained with the training sets. For the isolated gesture recognition task, we used the forward score $P(X|\lambda)$ for each sample X to choose a model with the highest likelihood. The result is given in Table IV. The recognition rate is the percentage of correctly recognized gestures from the number of all samples

$$\text{Recognition rate} = \frac{\#\text{ of correctly recognized gestures}}{\#\text{ of input gestures}} \times 100\%. \quad (18)$$

Since the test data and the models are different and the transition gesture model complicates segmentation, it is not possible to make a direct comparison between Tables II and IV. However, well-performing models in the isolated recognition task are more likely to perform better depending on the nature of individual gesture patterns, such as structural complexity, temporal duration, and so on. Note that the two gestures of "sitting on the floor" and "getting down on the floor" which perform poorly in

TABLE IV
RECOGNITION RATE OF ISOLATED GESTURES

Gestures	*N*	N_C	*R(%)*
Walking	50	49	98%
Running	50	48	96%
Bending	50	50	100%
Jumping	50	49	98%
Lying down on the floor	50	50	100%
Waving a hand	50	50	100%
Sitting on the floor	50	45	90%
Raising a right hand	50	50	100%
Getting down on the floor	50	46	92%
Touching a knee and wrist	50	50	100%
Total	500	487	97.4%

N : Number of input gestures
N_C : Number of correctly recognized gestures
R : Reliability

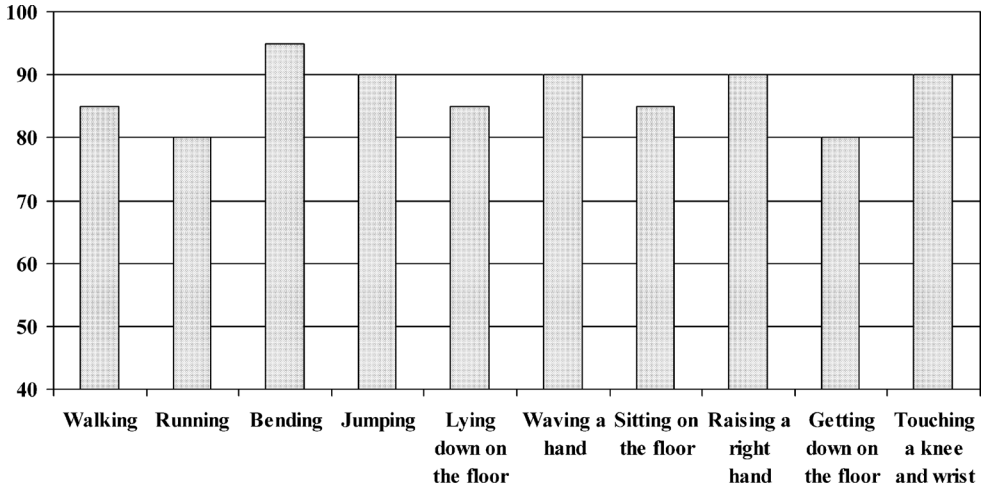

Fig. 23. Gesture spotting results with real-time data.

the isolated recognition task also worked poorly in the spotting task. If we improve those models in isolated gesture tasks, we believe that a further increase can be obtained in gesture spotting tasks.

D. Experimental Results With Real-Time Data

The first step in gesture video processing is to detect a human subject in each frame sequence. Then the 3-D pose of the human body is estimated using the pose reconstruction method described in [32]. It is based on a 3-D model shown in Fig. 22. Given an image captured with a stereo camera mounted on T-Rot, we identify individual body components. Fig. 22 shows various examples of 3-D component detection result in terms of 3-D models. The detected 3-D human body components are used to extract the features described in Section II-A. The human was extracted using the background subtraction method for real-time data. In this paper, research is focused on gesture spotting and recognition, which is a difficult task. Therefore, the human detection method was not considered.

Fig. 23 represents the result of gesture spotting and recognition with real-time data. As shown in the results, the gesture spotting results using real-time data is lower than the results using the KU Gesture DB. The average spotting result is about 87% with real-time data. The detection errors of 3-D human body components affect the result of gesture spotting. These results show that our method is also efficient for real-time data.

VI. Conclusions and Further Research

This paper proposed an HMM-based method of spotting and recognizing gestures embedded in continuous whole-body motion for HRI. The proposed method employs GMM clustering in the feature space, producing efficient transition gesture models. Feature space clustering and the transition gesture HMM state reduction together form a highly efficient recognition network. The method of merging two states based on relative entropy and data-dependent weighting allows the model to be more effective at capturing the variability in intergesture patterns. In fact, when compared with a recently proposed method, operating without explicit transition gesture modeling, a definite advantage was seen. In effect, the proposed transition gesture modeling is believed to be an excellent mechanism for recognizing gestures, as opposed to transition gestures, and rejecting these transition gestures.

This paper demonstrated that the proposed gesture recognition interface transcends to a much broader range of personal service robots. Near-term future work includes extending the proposed method for spotting and recognition of command gestures for HRI.

References

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

[2] J. Barbic, N. S. Pollard, J. K. Hodgins, C. Faloutsos, J. Y. Pan, and A. Safonova, "Segmenting motion capture data into distinct behaviors," in *Proc. Int. Conf. Graphics Interface*, 2004, pp. 17–19.

[3] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[4] A. Bruce and G. Gordon, "Better motion prediction for people tracking," in *Proc. Int. Conf. Robot. Autom.*, New Orleans, LA, Apr. 2004, pp. 1418–1423.

[5] J. A. De Preez, "Efficient training of high-order hidden Markov models using first-order representations," *Comput. Speech Lang.*, vol. 12, pp. 23–39, 1998.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[7] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robot. Auton. Syst.*, vol. 42, pp. 143–166, 2003.

[8] B.-W. Hwang, S. Kim, and S.-W. Lee, "2D and 3D full-body gesture database for analyzing daily human gestures," in *Advances in Intelligent Computing*. New York: Springer, 2005, vol. 3644, Lecture Notes in Computer Science, pp. 611–620.

[9] K. Kahol, "Gesture segmentation in complex motion sequences," Master's thesis, Arizona State Univ., Tempe, AZ, 2003.

[10] K. Kahol, P. Tripath, and S. Panchanthan, "Automated gesture segmentation from dance sequences," in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recog.*, Seoul, Korea, 2004, pp. 883–888.

[11] ——, "Documenting motion sequences: Development of a personalized annotation system," *IEEE Multimedia Mag.*, vol. 13, no. , pp. 35–47, 2006.

[12] D. Kortenkamp, E. Huber, and R. P. Bonasso, "Recognizing and interpreting gestures on a mobile robot," in *Proc. Amer. Conf. Artif. Intell.*, Portland, OR, Aug. 1996, pp. 915–921.

[13] H.-K. Lee and J. H. Kim, "An HMM-based threshold model approach for gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 961–973, Oct. 1999.

[14] Y. Nakauchi and R. Simmons, "A social robot that stands in line," *Auton. Robots*, vol. 12, no. 3, pp. 313–324, 2002.
[15] K. Nickel and R. Stiefelhagen, "Real-time person tracking and pointing gesture recognition for human-robot interaction," in *Computer Vision in Human-Computer Interaction*. New York: Springer-Verlag, 2004, vol. 3058, Lecture Notes in Computer Science, pp. 28–38.
[16] V. Pavlovic, R. Sharma, and T. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, Jul. 1997.
[17] D. Perzanowski, D. Brock, S. Blisard, W. Adams, M. Bugajska, A. Schultz, G. Trafton, and M. Skubic, "Finding the FOO: A pilot study for a multimodal interface," in *Proc. IEEE Syst., Man, Cybern. Conf.*, Washington, DC, Oct. 2003, pp. 3218–3223.
[18] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Robot. Auton. Syst.*, vol. 42, pp. 271–281, 2003.
[19] 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.
[20] B. Rime and L. Schiaratura, *Fundamentals of Nonverbal Behavior*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
[21] T. Starner, J. Weaver, and A. Pentland, "Real-time American Sign Language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, Dec. 1998.
[22] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. M. Buhmann, "Topology free hidden Markov models: Application to background modeling," in *Proc. Int. Conf. Comput. Vis.*, Vancouver, BC, Canada, Jul. 2001, pp. 294–301.
[23] R. Stiefelhagen, C. Fügen, P. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel, "Natural human-robot interaction using speech, gaze and gestures," in *Proc. Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 2422–2427.
[24] K. Takahashi, S. Seki, and R. Oka, "Spotting recognition of human gestures from motion images," (in Japanese) Inst. Electron., Inf. Commun. Eng., Japan, Tech. Rep. IE92-134, 1992.
[25] S. Thrun, "Toward a framework for human-robot interaction," *Human-Computer Interaction*, vol. 19, no. 1, pp. 9–24, 2004.
[26] T-Rot: Thinking Robot KIST, Center for Intelligent Robotics [Online]. Available: http://www.irobotics.re.kr
[27] United Nations and the International Federation of Robotics: World Robotics 2002. New York/Geneva, 2002.
[28] T. Varga and H. Bunke, "Generation of synthetic training data for an HMM-based handwriting recognition system," in *Proc. 7th Int. Conf. Document Anal. Recog.*, Edinburgh, U.K., Aug. 2003, pp. 618–622.
[29] A. J. Viterbi, "Error bounds for convolution codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Feb. 1967.
[30] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Auton. Robots*, vol. 9, no. 2, pp. 151–173, 2000.
[31] S. Waldherr, "Gesture recognition on a mobile robot," Master's thesis, Carnegie Mellon Univ., Pittsburgh, PA, 1998.
[32] H.-D. Yang, S.-K. Park, and S.-W. Lee, "Reconstruction of 3D human body pose for gait recognition," in *Biometrics*. New York: Springer-Verlag, 2006, vol. 3832, Lecture Notes in Computer Science, pp. 619–625.

Hee-Deok Yang received the B.S. degree in computer science from Chungnam National University, Daejeon, Korea, in 1998, and the M.S. degree in computer science and engineering from Korea University, Seoul, Korea, in 2003. Currently, he is working toward the Ph.D. degree in the Department of Computer Science and Engineering, Korea University.

His research interests include sign language recognition, gesture recognition, and face recognition.

A-Yeon Park received the B.S. degree in computer science from Sookmyung Women's University, Seoul, Korea, in 2003, and the M.S. degree in computer science and engineering from Korea University, Seoul, Korea, in 2005.

She is currently with LS Industrial Systems Corporation, Seoul, Korea. Her research interests include gesture recognition and human behavior analysis.

Seong-Whan Lee (SM'96) received the B.S. degree in computer science and statistics from Seoul National University, Seoul, Korea, in 1984, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1986 and 1989, respectively.

From February 1989 to February 1995, he was an Assistant Professor in the Department of Computer Science, Chungbuk National University, Cheongju, Korea. In March 1995, he joined the faculty of the Department of Computer Science and Engineering, Korea University, Seoul, Korea, where he is now a full Professor. He was a Visiting Professor at the Artificial Intelligence Laboratory, MIT, in 2001. He has more than 200 publications on computer vision and pattern recognition in international journals and conference proceedings, and authored 10 books. His research interests include computer vision, pattern recognition, and neural networks.

Dr. Lee was the winner of the Annual Best Paper Award of the Korea Information Science Society in 1986. He obtained the Outstanding Young Researcher Paper Award at the 2nd International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Professor Award from Chungbuk National University in 1994. He obtained the Outstanding Research Award from the Korea Information Science Society in 1996. He also received an Honorable Mention from the Annual Pattern Recognition Society for an outstanding contribution to the *Pattern Recognition Journal* in 1998. He is a Fellow of International Association for Pattern Recognition, a Senior Member of the IEEE Computer Society and a Life Member of the Korea Information Science Society.