PAPER

Recognition of Shape-Changing Hand Gestures

Mun-Ho JEONG[†], *Nonmember*, **Yoshinori KUNO**[††], **Nobutaka SHIMADA**[†],
and **Yoshiaki SHIRAI**[†], *Regular Members*

SUMMARY We present a method to track and recognize shape-changing hand gestures simultaneously. The switching linear model using active contour model well corresponds to temporal shapes and motions of hands. However, inference in the switching linear model is computationally intractable, and therefore the learning process cannot be performed via the exact EM(Expectation Maximization) algorithm. Thus, we present an approximate EM algorithm using a collapsing method in which some Gaussians are merged into a single Gaussian. Tracking is performed through the forward algorithm based on Kalman filtering and the collapsing method. We also present a *regularized smoothing*, which plays a role of reducing jump changes between the training sequences of shape vectors representing complex-variable hand shapes. The recognition process is performed by the selection of a model with the maximum likelihood from some trained models while tracking is being performed. Experiments for several shape-changing hand gestures are demonstrated.

key words: *gesture recognition, active contour, switching linear model*

1. Introduction

Gesture recognition plays an important role in a host of man-machine interaction applications. A well-known method in gesture recognition is HMM (Hidden Markov Model) [21], [24], [25], which is essentially a quantization of time series (observation sequence) into a small number of discrete states with transition probabilities between states. Most of schemes in gesture recognition are based on measurement spaces like HMM [3], [16].

Although these showed successful results, there are two bottlenecks. First, based on the distributions of independent measurements or observations, they have a limitation in dealing with time series having dependencies. Second, they also have difficulties when it is hard to measure the required information for recognition. Taking an instance of hand gestures, measurement-based schemes might have no problem in the case of using just positions or velocities of hands, which are easy to measure. However, such measurements are not enough to represent complex hand gestures. We change hand shapes while moving the hands in many cases. Measuring the outlines of shape-changing hands is not always feasible, especially under complicated backgrounds.

We propose to introduce a dynamic process to explain dependencies between spatio-temporal configurations of a gesture sequence. This helps to solve not only the first problem but the second. In fact, if a hand motion is known in advance, that is, a dynamic model of a hand is known, we might be able to infer the positions and shapes of the hand over time even if they cannot be completely measured. However, real cases are not so simple since hand gestures exhibit complex and rich dynamic behaviors.

A promising approach to representing complex dynamics is to adopt switching linear model, which consists of a few linear dynamic models with Markov switching between them, rather than a single linear dynamic model. A well-known problem in switching linear model, however, is that the presence of Markov switching makes exact inference impossible. In this paper, we use approximate inference based on a collapsing method to avoid the problem. Spatio-temporal estimations of shapes and positions of a hand, are performed by the collapsing method and Kalman filtering. To estimate the parameters of switching linear model, we present an EM learning process into which approximate inference using the collapsing method is well incorporated.

We adopt an active contour model using B-spline [2] to represent complex hand shapes. Hand contours are parameterized into shape vectors, and the shape vectors and their derivatives are considered as state vectors in the switching linear model. For learning of the model, it is necessary to collect training sequences of shape vectors. When shape-changing hand gestures are considered, even though outlines of a hand vary gradually over time, there often happen abrupt changes between the shape vectors on the shape space due to separate parameterizations. Suppose the state in which the index finger and the middle finger are touching. This state changes to a completely different state with the separating fingers only by a small motion of the fingers. This fact often leads to poor learning or makes initial tracking impossible in the EM learning process. In this paper, we propose a *regularized smoothing* method to solve this problem. It can

Manuscript received April 11, 2001.
Manuscript revised November 30, 2001.

[†]The authors are with the Department of Mechanical Engineering for Computer-Controlled Machinery, Osaka University, Suita-shi, 565–0871 Japan.

[††]The author is with the Department of Information and Computer Sciences, Faculty of Engineering, Saitama University, Saitama-shi, 338–8570 Japan.

make a training sequence of shape vectors vary gradually but the outlines of the hand remain invariant with allowable errors.

Our previous study concerned hand posture estimation using outlines of a hand, but it makes black background a condition for perfect retrieving outlines of a hand [22]. The first efforts at classification and tracking of hand gestures using active contour model and multiple dynamic models were made by Isard et al. [10]. They showed tracking of hand outlines and classification of different writing patterns. However, they confined the scope of changes in hand shapes to affine transformation. Pavlovic and Rehg applied switching linear model to tracking of human figures [18]. They used Viterbi approximation to overcome the exponential complexity of exact inference. The above approaches did not handle online recognition during tracking, but concentrated on tracking of a human gesture or the problem of where to switch to another dynamics in time domain. Pentland and Liu modeled automobile drivers' actions by Hidden Markov Dynamic Model in which Kalman filtering method is incorporated into HMM structure [17]. In their learning process, estimation of dynamic parameters was not incorporated into EM learning (Baum-Welch algorithm).

Recognition process in shape-changing hand gestures is to choose the switching linear model which gives the best tracking result. We propose a method of tracking shape-changing hand gestures and recognizing them by online selection of an appropriate model.

The paper is organized as follows. In the following section we address the switching linear model. The forward and the backward algorithms for approximate inference are explained. In Sect. 3, we concern tracking implementation where the difficulties in applying an active contour model to shape-changing hand gestures are addressed. In Sect. 4, we present a *regularized smoothing* method and explain the EM learning using a collapsing method for the switching linear model. In Sect. 5, we address the recognition process where computation of likelihood given a model is explained. The experimental results are shown in Sect. 6. Finally, we conclude with Sect. 7.

2. Switching Linear Model

2.1 Model Specification

Switching linear model can be seen as a hybrid model of the linear state-space model and HMM. It is described using the following set of state-space equations:

$$\begin{aligned} & x_t = F_{m_t}x_{t-1} + D_{m_t} + u_t, \quad u_t \sim N(0, Q_{m_t}) \\ & \varPhi_{m_t,m_t+1} = p(m_t+1|m_t) \\ & \pi_{m_1} = p(m_1), \quad m_t \in \{1, 2, \cdots, M\} \end{aligned} \tag{1}$$

In the above equations, x_t is a hidden continuous state vector. u_t is independently distributed on the Gaussian distribution with zero-mean and covariance Q_{m_t}. π_{m_1}, F_{m_t}, and D_{m_t}, which are typical parameters of linear dynamic model, denote the prior probability of a discrete state, the continuous state transition matrix, and the offset, respectively. The parameters with the subscript m_t are dependent on the discrete state variable m_t indexing a linear dynamic model. And the switching process between M discrete states obeys the first Markov process and is defined with the discrete state transition matrix $\varPhi$. Hence it follows that

$$p(m_t|m_1, \cdots, m_{t-1}) = p(m_t|m_{t-1}). \tag{2}$$

This model can be illustrated in Fig. 1 where o_t denotes an observation vector at time t. It is assumed that o_t is statistically independent from all other observation vectors.

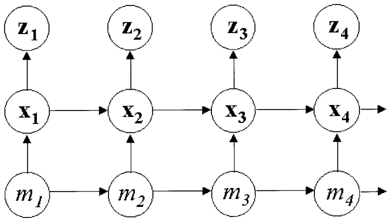


Fig. 1 Graphical model of switching linear model. Arrows denote probabilistic dependencies.

2.2 Forward Algorithm

Given known parameters of the switching linear model, $\{F_j, D_j, Q_j, \pi_j, \varPhi | j = 1, 2, \cdots, M\}$, we can perform tracking or filtering, which means estimations of continuous states and probabilities of joint-discrete states here. The predicted joint-continuous state vector and its covariance are derived dependently on $m_{t-1} = i$ and $m_t = j$:

$$\begin{aligned} x^{(i,j)}_{t|t-1} &= F_j x^{(i)}_{t-1|t-1} + D_j \\ P^{(i,j)}_{t|t-1} &= F_j P^{(i)}_{t-1|t-1} F_j^{'} + Q_j \end{aligned} \tag{3}$$

where $x^{(i)}_{t-1|t-1}$ and $P^{(i)}_{t-1|t-1}$ are estimations at time $t-1$ on the condition given observations up to time $t-1$. Now the filtered joint-continuous state and its covariance, $x^{(i,j)}_{t|t}$, $P^{(i,j)}_{t|t}$, are estimated by the conventional Kalman filtering method. In particular, we follow Kalman filtering application to active contour model by Blake [1], [2].

From the above fact, as noted by Gordon and Smith [6], switching linear dynamic model requires computing a Gaussian mixture with M^t components at time t for M switching states. That leads to intractable inference for moderate sequence length. It is necessary to introduce some approximations to solve the intractable computation problem.

We collapse M^2 joint-continuous state vectors into M state vectors at each time to prevent M-fold increase in the number of cases to consider. Thus we can avoid prohibitive increase of computational cost. For example, given $M = 3, t = 20$, the number of cases to consider amounts to $3,486,784,401$ for exact inference, whereas if collapsing is used, it needs just 3^2 computations regardless of t. Building upon the ideas introduced by Harrison [9], Gordon [6] and Kim [12], the collapsing is given by

$$x_{t|t}^{(j)} = \frac{\sum_{i=1}^{M} p(m_{t-1}=i, m_t=j|O_t)\cdot x_{t|t}^{(i,j)}}{p(m_t=j|O_t)}$$

$$P_{t|t}^{(j)} = \frac{\sum_{i=1}^{M}\left(\begin{smallmatrix} p(m_{t-1}=i,m_t=j|O_t)\cdot \\ \left(P_{t|t}^{(i,j)}+(x_{t|t}^{(j)}-x_{t|t}^{(i,j)})(x_{t|t}^{(j)}-x_{t|t}^{(i,j)})'\right)\end{smallmatrix}\right)}{p(m_t=j|O_t)} \tag{4}$$

where O_t is an observation sequence $\{o_1, o_2, \cdots, o_t\}$ and o_t is an observation vector. In the above collapsing, the probabilities of joint-discrete states play a role of weighting factors of joint-continuous state vectors. To complete the collapsing, we have only to calculate the weighting factors.

The probabilities of the filtered joint-discrete states are approximately obtained by (refer to Appendix B)

$$\begin{aligned} &p(m_{t-1}, m_t|O_t) \\ &\quad = k_t p(m_{t-1}, m_t|O_{t-1}) p(o_t|x_{t|t-1}^{(m_{t-1},m_t)}) \end{aligned} \tag{5}$$

where k_t is a normalizing constant. From (2), the prediction step given sequence up to time t gives

$$\begin{aligned} &p(m_t, m_{t+1}|O_t) \\ &\quad = \Phi_{m_t,m_{t+1}} \sum_{m_{t-1}=1}^{M} p(m_{t-1}, m_t|O_t). \end{aligned} \tag{6}$$

Then the probabilities of the predicted and the filtered discrete states are calculated by the following marginalization.

$$\begin{aligned} p(m_t|O_t) &= \sum_{m_{t+1}=1}^{M} p(m_t, m_{t+1}|O_t) \\ p(m_{t+1}|O_t) &= \sum_{m_t=1}^{M} p(m_t, m_{t+1}|O_t) \end{aligned} \tag{7}$$

We can estimate the filtered continuous state vector by taking a weighted average over the discrete states at time t from

$$x_{t|t} = \sum_{j=1}^{M} p(m_t=j|O_t) x_{t|t}^{(j)}. \tag{8}$$

2.3 Backward Algorithm

While the forward algorithm is a filtering process given sequence up to current time, the backward algorithm is a smoothing process given sequence of full length. Like the conventional Kalman smoothing method, the joint-continuous state vector and its covariance based on full sequence can be smoothed as follows: given $m_t = j$ and $m_{t+1} = k$,

$$\begin{aligned} x_{t|T}^{(j,k)} &= x_{t|t}^{(j)} + \tilde{P}_t^{(j,k)}(x_{t+1|T}^{(k)} - x_{t+1|t}^{(j,k)}) \\ P_{t|T}^{(j,k)} &= P_{t|t}^{(j)} + \tilde{P}_t^{(j,k)}(P_{t+1|T}^{(k)} - P_{t+1|t}^{(j,k)})\tilde{P}_t^{\prime(j,k)} \end{aligned} \tag{9}$$

where $\tilde{P}_t^{(j,k)} = P_{t|t}^{(j)} F_k'(P_{t+1|t}^{(j,k)})^{-1}$. To calculate the smoothed continuous state vector and its covariance, collapsing is performed similarly to (4):

$$x_{t|T}^{(j)} = \frac{\sum_{k=1}^{M} p(m_t=j, m_{t+1}=k|O_T)\cdot x_{t|T}^{(j,k)}}{p(m_t=j|O_T)}$$

$$P_{t|t}^{(j)} = \frac{\sum_{k=1}^{M}\left(\begin{smallmatrix} p(m_t=j,m_{t+1}=k|O_T) \\ \cdot\left(P_{t|T}^{(j,k)}+(x_{t|T}^{(j)}-x_{t|T}^{(j,k)})(x_{t|T}^{(j)}-x_{t|T}^{(j,k)})'\right)\end{smallmatrix}\right)}{p(m_t=j|O_T).} \tag{10}$$

To complete (10), we turn to derivation of the probabilities of the smoothed joint-discrete states, which is given by

$$p(m_t, m_{t+1}|O_T) = p(m_t, m_{t+1}|O_t)\frac{p(m_{t+1}|O_T)}{p(m_{t+1}|O_t)}. \tag{11}$$

From (11) the probabilities of the smoothed discrete states is obtained as

$$p(m_t|O_T) = \sum_{m_{t+1}=1}^{M} p(m_t, m_{t+1}|O_T) \tag{12}$$

$p(m_T|O_T)$ and $p(m_{t+1}|O_t), t = 1, \cdots, T-1$, have already been computed from (7) in the forward algorithm.

3. Tracking Implementation

Estimation of an initial state is important in tracking problems because any tracking method can operate by its own scheme after the initial state was found. In fact, initial estimation is a problem of segmentation to extract tracking targets regardless of tracking itself. If hand gesture recognition is considered together with tracking, initial estimation problem becomes more difficult because we have to know when a gesture starts and finishes in addition to detecting a hand. To simplify the problem we assume that initial state has been known in advance as its mean and covariance and use gesture sequences of a specified length. We just concentrate on evolution of states in this paper. Detecting meaningful hand gestures from idle movements of a hand is left for our future work. However, since recognition in the case of connected cases has been well studied such as in [19], [24], the techniques used there can be extended for our method.

We intend to track hand motions including changes in hand shapes. To represent a variety of shapes of a hand, we follow the active contour model where outlines of the hand are parameterized into control vectors composed of B-spline control points [2]. A control vector is transformed to a low-dimensional shape vector on a specific shape space formed by PCA(principal component analysis) method.

Active contour models such as snakes and deformable templates have practical problems in being applied to tracking hand gestures. Although the schemes are effective to retrieve features with geometric structures, they are too sensitive to noises to track an object under a complicated background and also have difficulties in progressing into boundary concavities which are frequently seen in shape-changing hand gestures.

There have been dynamic contour methods using dynamic models as predictors in visual trackers [14]. However, the constant velocity Kalman tracker, which is popularly used for tracking, is too easily distracted by clutter of the background to track a hand against a rapidly changing motion as shown in Fig. 2 (a). In addition, the tracker often fails to track a hand motion causing boundary concavities even under a black background as shown in Fig. 2 (b). To tackle the problems, it is required to have a more precisely tuned predictors constructed by multiple dynamic models rather than a simple or single dynamic model.

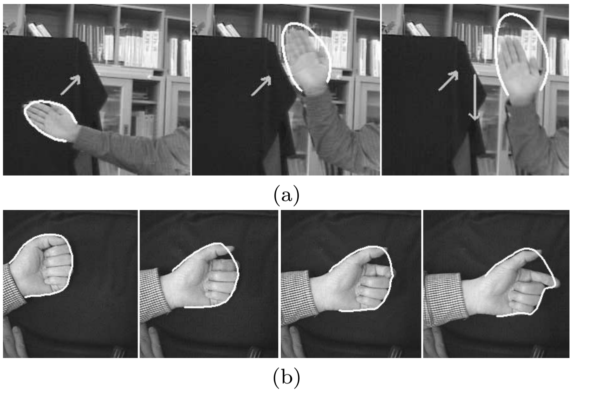
For the purpose of tracking complex hand gestures, we assume that temporal shapes and motions of the hand have dynamic behaviors characterized by switching linear dynamics. Then to incorporate the active contour model into the switching linear model, a state vector in (1) is defined as follows:

$$x_t = \begin{pmatrix} \mathrm{v_t} \\ \dot{\mathrm{v}}_\mathrm{t} \end{pmatrix} \tag{13}$$

where $\mathrm{v_t}$ is the shape vector representing the hand outline at t.

An observation(o_t) is shown as feature points r_{f_t} which are edges detected by line searching along the normal direction at sample points on a specific contour [2]. For robustness and accuracy of one dimensional edge detection, we use a Mahalanobis distance from a mean color of a hand with its covariance rather than common gray intensities. And we also extract edges with outward orientation on the assumption that Mahalanobis distances of interior parts of a hand are small and uniform.

Now an observation probability given a predicted state vector in (5) is computed by


(a)

(b)

Fig. 2 Constant velocity Kalman tracker. In (a) the Kalman tracker works well against the background clutter in smooth hand motions, but the tracker is distracted by the clutter when the hand moves downwards rapidly. In (b) the Kalman tracker often has difficulties in progressing into boundary concavities.

$$\begin{aligned} &p(o_t|x_{t|t-1}) = p(o_t|\mathrm{v_{t|t-1}}) \\ &\quad \propto \exp -\frac{d}{2\sigma^2 N}\sum_{i=1}^{N}[(r_{f_t}(s_i) - r_t(s_i)) \cdot \bar{n}(s_i)]^2 \end{aligned} \tag{14}$$

where σ^2 is an assumed variance due to measuremnt error, d is the dimension of shape vector and $\bar{n}(s_i)$ is a normal unit vector at $r_t(s_i)$ of N sample points on the predicted contour, $\mathrm{v_{t|t-1}}$.

4. Learning

4.1 Learning via EM Algorithm

EM algorithm is a general iterative technique for finding maximum likelihood parameter estimates in problems where some variables are unobserved [5]. In our case, continuous state variables and discrete state variables are unobserved. Assume that the probability density for observation sequence is parameterized using $\lambda = \{F_j, D_j, Q_j, \pi_j, \Phi | j = 1, 2, \cdots, M\}$, $p(O_T|\lambda)$, the log-likelihood is given by

$$\begin{aligned} &Likelihood(\lambda|O_T) \\ &\quad = \log p(O_T|\lambda) \\ &\quad = \log \sum_{M_T} \int_{X_T} p(M_T, X_T, O_T|\lambda) dX_T \end{aligned} \tag{15}$$

where M_T and X_T are sequences (of length T) of discrete states and continuous states, respectively. Neal and Hinton [15] showed that the auxiliary log-likelihood is given by

$$\begin{aligned} L &= \sum_{M_T} \int_{X_T} \begin{pmatrix} p(M_T, X_T|O_T, \bar{\lambda}) \\ \cdot \log p(M_T, X_T, O_T|\lambda) \end{pmatrix} dX_T \\ &= E_{\bar{p}}[\log p(M_T, X_T, O_T|\lambda)] \end{aligned} \tag{16}$$

where $\bar{p} = p(M_T, X_T|O_T, \bar{\lambda})$ and $\bar{\lambda}$ is previously estimated parameter set. From Fig. 1, the joint probability for the sequences of states X_T, M_T and observations O_T can be factored as:

$$\begin{aligned}
& p(M_T, X_T, O_T|\lambda) \\
&\quad = p(m_1)p(x_1|m_1)p(o_1|x_1, m_1)\prod_{t=2}^{T} p(o_t|x_t, m_t) \\
&\qquad \times \prod_{t=2}^{T} p(m_t|m_{t-1})p(x_t|x_{t-1}, m_{t-1}, m_t)) \\
&\quad = \pi_{m_1} p_{m_1}(x_1) p_{m_1}(o_1|x_1) \prod_{t=2}^{T} p_{m_t}(o_t|x_t) \\
&\qquad \times \prod_{t=2}^{T} \varPhi_{m_{t-1},m_t} p_{m_{t-1},m_t}(x_t|x_{t-1}) \\
&\quad = \prod_{j=1}^{M} [\pi_j p_j(x_1) p_j(o_1|x_1)]^{\psi_1(j)} \\
&\qquad \times \prod_{t=2}^{T}\prod_{j=1}^{M} [p_j(o_t|x_t)]^{\psi_t(j)} \\
&\qquad \times \prod_{t=2}^{T}\prod_{i=1}^{M}\prod_{j=1}^{M} [\varPhi_{i,j} p_{i,j}(x_t|x_{t-1})]^{\psi_{t-1}(i)\psi_t(j)} \quad (17)
\end{aligned}$$

where $\psi_t(k) = 1$ if $m_t = k$, otherwise 0. Based on the collapsing method in the presented switching linear model, it follows that

$$p_{i,j}(x_t|x_{t-1}) \simeq \frac{\sqrt{\det(Q_j^{-1})}}{\sqrt{2\pi}^d} \exp\left(\eta_t^{'(i,j)} Q_j^{-1} \eta_t^{(i,j)}\right) \quad (18)$$

where $\eta_t^{(i,j)} = (x_t^{(i,j)} - F_j x_{t-1}^{(i)} - D_j)$ and d is dimension of state vectors. Substituting (17) and (18) into (16), then L can be approximately obtained as the followings, up to constants: (refer to Appendix C)

$$\begin{aligned}
L \simeq \tilde{L} = &\sum_{t=2}^{T}\sum_{i,j=1}^{M} \begin{pmatrix} p(m_{t-1} = i, m_t = j|O_T)\cdot \\ \frac{1}{2}(\det(Q_j^{-1}) - \eta_{t|T}^{'(i,j)} Q_j^{-1} \eta_{t|T}^{(i,j)}) \end{pmatrix} \\
&+ \sum_{t=2}^{T}\sum_{i,j=1}^{M} p(m_{t-1} = i, m_t = j|O_T) \log \varPhi_{i,j} \\
&+ \sum_{i=1}^{M} p(m_1 = i|O_T) \log \pi_i \qquad (19)
\end{aligned}$$

EM algorithm starts with an initial guess of parameters and proceeds by applying the following two steps repeatedly:

E step On the condition given the observation sequence of full length and the previous parameter set, O_T, $\bar{\lambda}$, we estimate continuous states $x_{t|T}^{(m_t)}$, joint-continuous states $x_{t|T}^{(m_{t-1},m_t)}$, and probabilities of joint-discrete states and discrete states, $p(m_{t-1}, m_t|O_T)$ and $p(m_t|O_T)$. These estimations are performed through the forward and the backward processes described in Sects. 2.2 and 2.3.

M step If $\tilde{L}$ is expressed by λ and the estimations from E step, then we estimate λ maximizing $\tilde{L}$. The new parameter set, $\lambda = \{F_j, D_j, Q_j, \pi_j, \varPhi | j = 1, 2, \cdots, M\}$, is obtained as Appendix A.

The above two steps are iterated until the likelihood value converges. The likelihood value can be computed easily by (23) given in the following section.

4.2 Practical Problems

EM learning starts with an initial guess of parameters. If shape-changing hand gestures are considered as in the paper, initial guess of them become more significant because with a default or roughly determined dynamics it is impossible to track a shape-changing hand gesture. Initial tracking has to be feasible to some extent so that the dynamic model can be improved by iterative adjustment of dynamic parameters. Thus, with respect to each sequence segmented manually, initial estimation of parameters is performed by the maximum likelihood method.

It is necessary to prepare training sequences of state vectors. That is essential to general learning processes using the maximum likelihood method. Even though such a sequence can be obtained in various ways, there often occurs a problem in the case of shape-changing hand gestures considered here. Generally, outlines of a hand have gradual changes over time. However, shape vectors representing its outlines often vary abruptly on the shape space due to separate parameterizations of outlines of the hand. This often leads to poor learning.

In fact, if the training sequence can be obtained sequentially and entirely through an iterative contour fitting using B-spline snake or deformable templates, it might be possible to avoid the jump changes. That is, beginning from the initial outline of the hand, the fitted contour is projected onto the shape space with a gradual change from the previous shape vector since the current outline of the hand is fitted from the previous contour (shape vector) and the detected feature map. However, as addressed above, this process is not always successful because of boundary concavities.

4.3 Regularized Smoothing

In this section, we present the *regularized smoothing* method to make a training sequence of shape vectors have gradual changes on the shape space but outlines of the hand remain invariant with allowable errors. To reduce the undesirable jump changes between shape vectors as many as possible, we can apply B-spline fitting using deformable templates partially as long as it works

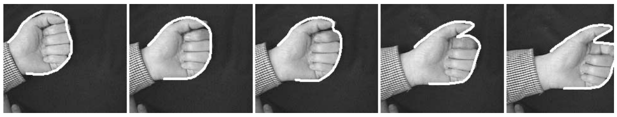


Fig. 3 An initially given training sequence of contours. These contours are originated from two differently parameterized contour. The length of the sequence and the size of an image is 30 frames and 360×240 pixels, respectively. From left to right, the frame numbers are 1,21,22,26 and 30.

well. Figure 3 shows a moving hand sequentially. Contours from the 1st frame to the 21st frame were obtained by iterative curve fitting from the first contour. The others were fitted backwardly from the last contour. As expected from the previous section, it was found that a jump exists between 21st frame and 22nd frame because they were originated from two separate parameterizations. Figure 4 shows the shape vectors representing the contour shape on the space of the first two dimensions of the PCA shape space. The black dots clearly show this jump.

Given a sequence of shape vectors, $\mathrm{v}_1^o, \mathrm{v}_2^o, \cdots, \mathrm{v}_T^o$, the new fitted shape vector at time t, v_t, can be obtained by the following *regularized smoothing.*

Step 1 Initialize $\mathrm{v}_t = \mathrm{v}_t^o$, $t = 1, 2, \cdots, T$

Step 2 Estimate

$$\mathrm{v}_t = \arg\min_{\hat{\mathrm{v}}_t} \alpha \left\| \hat{\mathrm{v}}_t - \frac{3\mathrm{v}_{t-1} + \mathrm{v}_{t+1} - \mathrm{v}_{t-2}}{3} \right\|^2 + \frac{1}{N}\sum_{i=1}^{N}[(\hat{r}_t(s_i) - r_t^o(s_i)) \cdot \bar{n}(s_i)]^2, \quad t = 3, 4, \cdots, T-1 \tag{20}$$

Step 3 Iterate Step 2

where α is a *regularized smoothing* constant, $\hat{r}_t$ and r_t^o are the contour points of $\hat{\mathrm{v}}_t$ and v_t^o, respectively, and $\| \; \|$ denotes L_2 norm [2]. $\bar{n}(s_i)$ is the normal vector at a sampled point of $\hat{r}_t$. The first part in (20), which describes a smoothness constraint using discrete time acceleration, forces the current state to be positioned for smooth changes in a local interval while the second part guarantees that the fitted curve remains unchanged. The *regularized smoothing* controls trade-off from smooth changes between shape vectors to more accurate fitting. A smoothness error and a contour error are defined as the first norm and the second part in (20) after the *regularized smoothing*, respectively. Figure 5 shows that the smoothness improves but the fitting error increases as the constant increases. We have found that the constant $\alpha = 2.5$ is adequate to the trade-off. In the case of $\alpha = 2.5$, the maximum contour error amounts to 5.37 pixel^2 at 22 nd frame. In Fig. 4, the smoothed sequence of shape vectors is shown as the white-dots curve without any jump.

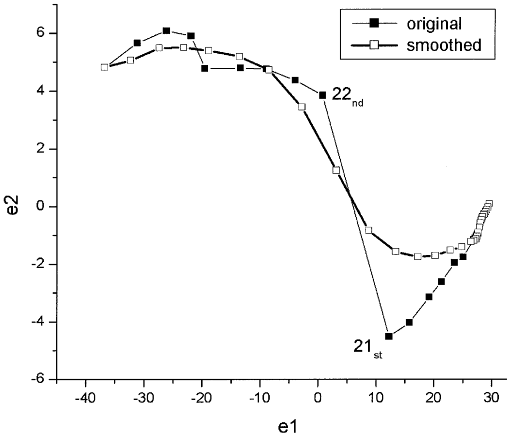


Fig. 4 The result of the *regularized smoothing.* The original sequence has a jump change between 21st and 22nd. The jump change disappears after the *regularized smoothing* with $\alpha = 2.5$.

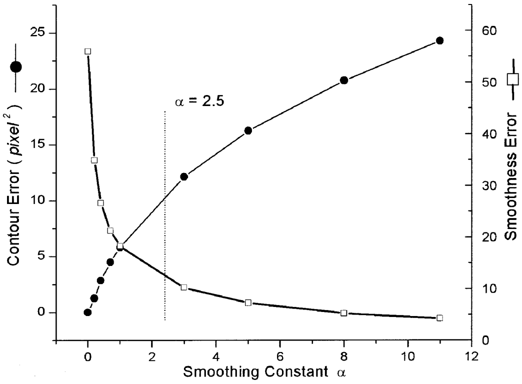


Fig. 5 Smoothness error vs. contour error. Total smoothness error is computed by summation of smoothness errors over time. Total contour error is computed by summation of contour errors over time likewise.

5. Recognition

Recognition of hand gestures can be considered as the problem to determine which model tracks a hand gesture well. Therefore, a given sequence of hand gestures can be recognized by means of the likelihood values of candidate models. As addressed in Sect. 1, our goal is to track and recognize hand gestures simultaneously. We have to compute the likelihood of each model while tracking is being performed by the forward algorithm. Then, a given sequence of hand gestures can be recognized as the model with the maximum likelihood.

The switching linear model can be represented by the parameter set λ. The likelihood of λ given an observation sequence can be calculated by

$$L(\lambda | O_\mathcal{T}) = p(O_\mathcal{T} | \lambda) = \prod_{t=1}^{\mathcal{T}} p(o_t | O_{t-1}, \lambda). \tag{21}$$

Abbreviating λ, from (A· 2) and (A· 4)

$$p(o_t|O_{t-1}) = \frac{1}{k_t} \tag{22}$$

Substituting this into (21), log-likelihood L_τ is obtained by

$$L_\tau = \sum_{t=1}^{\tau} \log\left(\frac{1}{k_t}\right) \tag{23}$$

where k_t has been computed in the forward algorithm.

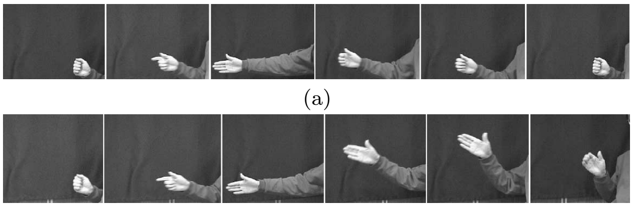
6. Experimental Result

We used image sequences with black backgrounds for training. A PCA shape space of the hand was built with control vectors of contours estimated or corrected manually from image sequences. Then we applied the *regularized smoothing* to the shape vectors projected onto the shape space. The smoothed training sequence is segmented by hand to distinguish linear models indexed by discrete states. Initial estimation of each linear model is performed with each segmented sequence by the maximum likelihood method. Then EM leaining using the collapsing method was performed with respect to the smoothed training sequence. We applied the trained model to a test image sequence with a complicated background. We used different variance of measurement error, σ, of (14) in a test and training ($\sigma_{\text{test}} : 4_{\text{pixel}}, \sigma_{\text{training}} : 2_{\text{pixel}}$). It is necessary to increase for a complicated background. σ is used as a factor in Kalman updating of shape vectors besides (14) [2].

We constructed two gesture models, A and B. Gesture model A represents a gesture composed of spreading and folding a hand. Gesture model B describes a gesture in which one stretches his arm while his fist is unfolded and clenches his fist following clockwise movement of the hand. The number of discrete states of a gesture model was determined as a minimum number enough to track the gesture. The gesture models A and B were designed to have two and three discrete states, respectively. Figure 6 shows the training sequence of each gesture model.

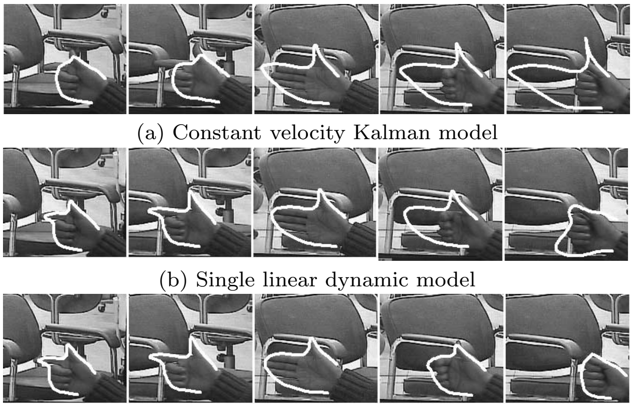
Figure 7 shows tracking results by conventional trackers and the new tracker using the presented scheme. In Fig. 7 (a) the constant velocity Kalman tracker tracked the hand roughly but did not catch severe concavities caused by the index finger. The tracker with the single linear dynamic model trained for tracking severe changes in the hand shapes was distracted when the hand moves rapidly as shown in Fig. 7 (b). However, the presented tracker using gesture model A tracked the hand gesture successfully as shown in Fig. 7 (c). Figure 8 shows tracked contours by model B. Transition between the discrete states is described in Fig. 9.

To show recognition during tracking, we applied two models to each test sequence as shown in Fig. 10. The likelihood of each model is computed at every time. Recognition is performed by selection of a model with the largest likelihood. In the experiment in Fig. 10 (b), the likelihood of model A decreases sharply since the tracker of model A is distracted because of the rapid upward-movement of the hand as plotted in Fig. 11.


(a)

(b)

Fig. 6 Training image sequences for two gesture models. In (a), the training image sequence of gesture model A is shown. Discrete state 1 corresponds to spreading a hand through shape changes: *stone-scissors-paper*(1st, 2 nd and 3 rd pictures from the leftmost). The motion of folding the hand is represented by discrete state 2(3 rd, 4 th, 5 th and 6 th pictures from the leftmost). (b) shows the training sequence of gesture model B. In discrete state 1 one's arm is being stretched while his fist is unfolded (1st, 2 nd and 3 rd pictures from the leftmost). Discrete state 2 corresponds to clockwise movement of the hand (3 rd, 4 th and 5 th pictures from the leftmost). Discrete state 3 represents clenching one's fist(the rightmost picture).


(a) Constant velocity Kalman model

(b) Single linear dynamic model

(c) Proposed gesture model

Fig. 7 Tracking examples.

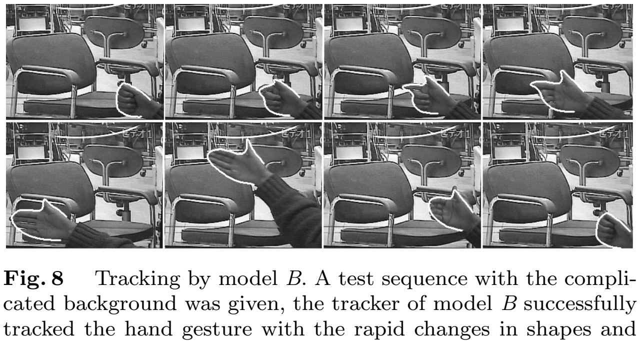


Fig. 8 Tracking by model B. A test sequence with the complicated background was given, the tracker of model B successfully tracked the hand gesture with the rapid changes in shapes and positions. From left-top to right-down, the frame numbers are 1,21,25,31,46,77,92 and 105, respectively.

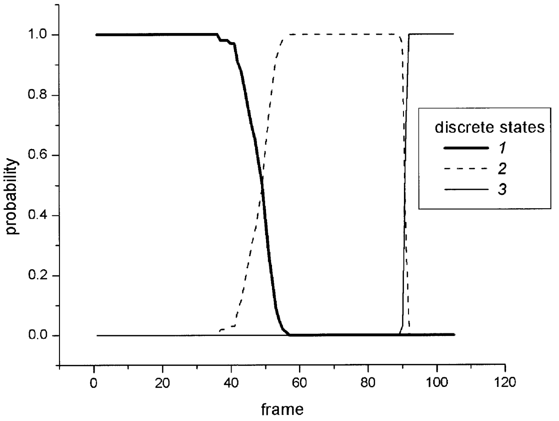


Fig. 9 Transition between discrete states.

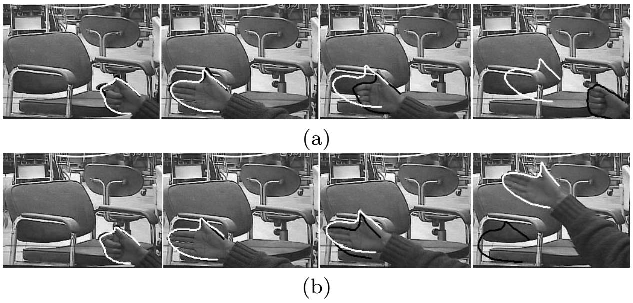


Fig. 10 Tracking and recognition. The black contour is the tracker of model *A* and the white contour is the tracker of model *B*. Two trackers start tracking from the known initial state, but only the tracker of model *A* works correctly in (a) while the tracker of model *B* in (b). Recognition is performed during tracking by comparision of likelihood at every time.

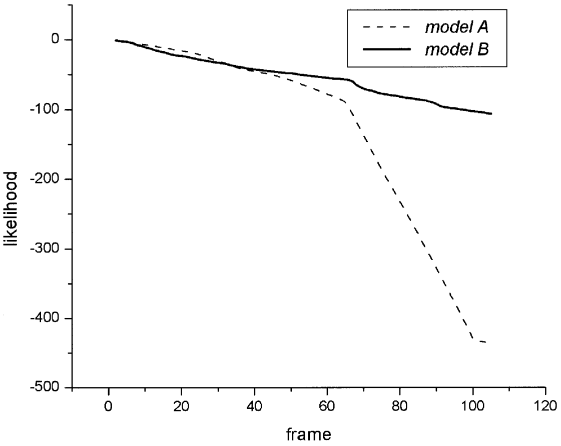


Fig. 11 Likelihood for recognition. This figure shows likelihood of each model at each time in the case of Fig. 10 (b). Difference between likelihood values of two models increases rapidly since the tracker of model *A* is distracted.

7. Conclusion

We have presented a framework to track and recognize shape-changing hand gestures simultaneously. To model complex and rich dynamic behaviors of hands, we have introduced switching linear model in which shape vectors, which are parameterizations of hand contours by active contour model, are considered as state vectors. To overcome exponential complexity of exact inference in switching linear model, an approximate inference is performed by a collapsing method in which some Gaussian distributions of state vectors are merged.

The parameters of the model are estimated via EM algorithm into which the collapsing method is incorporated. We have also presented a smoothing method using regularization for smoothness in a training sequence of shape vectors. Through this process, we obtain shape vectors shifted on a shape space while real outlines of the hand remain invariant with allowable errors.

Recognition is performed by selection of a model out of some trained models through log-likelihood values of each model. Log-likelihood values are computed from a forward algorithm that performs filtering process based on the collapsing method. Thus, we can recognize shape-changing gestures during tracking. Experimental results show that shape-changing hand gestures are recognized and tracked simultaneously using the presented scheme.

Although we achieved satisfactory results in tracking under complicated background, there still remains a problem that the allowable error bound in the approximate inference is not known. In other words, we have no information enough to make sure that the collapsing method can cope with all various backgrounds. As an alternative, there are Monte-Carlo-based methods in which a number of samples are used to represent the probability densities of state vectors [7], [11], [13]. Although this approach is plausible to complex-cluttered backgrounds, this is often considered not to be feasible for real time applications because the exponential number of samples is required for high dimensional space, which is general in shape-changing hand gestures.

The presented scheme is expected to be applicable to the recognition of sign languages. We are planning to examine usefulness of our scheme for sign language system. This is left for our future work.

References

[1] A. Blake, M. Isard, and D. Rubin, "Learning to track the visual motion of contour," Some Artificial Intelligenc, vol.78, pp.101–134, 1995.

[2] A. Blake and M. Isard, Active contour, Springer-Verlag, London, 1998.

[3] A.F. Bobick and A.D. Wilson, "A state-based approach to the representation and recognition of gesture," IEEE Trans. Pattern Anal. & Mach. Intell., vol.19, pp.1325–1337, 1997.

[4] W.S. Chaer, R.H. Bishop, and J. Ghosh, "A mixture-of-experts framework for adaptive Kalman filtering," IEEE Trans. Syst., Man. & Cybern., 1997.

[5] A. Dempster, M. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Royal

Statistical Society, vol.B39, pp.1–38, 1977.
[6] K. Gordon and A. Smith, “Modeling and monitoring discontinuous changes in time series,” in Bayesian Analysis of Time Series and Dynamic Linear Model, ed. M. Dekker, pp.359–392, New York, NY, 1988.
[7] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” IEE Proceedings-F, vol.140, no.2, pp.107–113, 1993.
[8] Z. Ghahramani and G.E. Hinton, “Variational learning for switching state-space models,” CRG-TR-96-3 of Toronto Univ., 1996.
[9] P.J. Harrison and C.F. Stevens, “Bayesian forecasting,” J. Royal Statistical Society, vol.B38, pp.205–247, 1977.
[10] M. Isard and A. Blake, “A mixed-state condensation tracker with automatic model switching,” Proc. 6th Int. Conf. Computer Vision, pp.107–112, 1998.
[11] M. Isard and A. Blake, “Condensation-conditional density propagation for visual tracking,” Int. J. Computer Vision, vol.29, no.1, pp.5–28, 1998.
[12] C.-J. Kim, “Dynamic linear models with Markov-switching,” J. Econometrics, vol.60, pp.1–22, 1994.
[13] G. Kitagawa, “Mont Carlo filter and smoother for non-Gaussian nonlinear state space models,” J. Computational and Graphical Statistics, vol.5, no.1, pp.1–25, 1996.
[14] D. Metaxas and D. Terzopoulos, “Constrained deformable superquadrics and nonrigid motion tracking,” Proc. CVPR, pp.337–343, 1991.
[15] R.M. Neal and G.E. Hinton, “A view of the EM algorithm that justifies incremental, sparse and other variants,” Learning in Graphical Model, pp.355–368, Kluwer Academic Publishers, Dordrecht, 1998.
[16] H. Ohno and M. Yamamoto, “Gesture recognition using character recognition techniques on two-dimensional eigenspace,” Int. Con. Automatic Face and Gesture Recognition, pp.151–156, 1999.
[17] A. Pentland and A. Liu, “Modeling and prediction of human behavior,” Technical Reports 433, MIT Media Lab., 1995.
[18] V. Pavlovic and J. Rehg, “A dynamic Bayesian network approach to figure tracking using learned dynamic models,” Int. Con. Automatic Face and Gesture Recognition, pp.94–101, 1999.
[19] L.R. Rabiner, “A tutorial on hidden Markov models and seledted applications in speech recognition,” Proc. IEEE, vol.77, no.2, Feb. 1989.
[20] Y. Sakamoto, M. Ishiguro, and G. Kitagawa, Akaike Information Criterion Statistics, D.Reidel, Tokyo, 1986.
[21] J. Schlenzig, E. Hunter, and R. Jain, “Recursive identification of gesture inputers using hidden Markov models,” Proc. Second Ann. Conf. on Appl. of Computer Vision, pp.187–194, 1994.
[22] N. Shimada, K. Kimura, Y. Shirai, and Y. Kuno, “Hand posture estimation by combining 2-D appearance-based and 3-D model-based approaches,” ICPR, pp.709–712, 2000.
[23] R.H. Shumway and D.S. Stoffer, “An approach to time series smoothing and forecasting using the EM algorithm,” J. Time Series Analysis, vol.3, no.4, pp.253–264, 1982.
[24] T. Starner and A. Pentland, “Real time American sign language recognition from video using hidden Markov models,” Technical Report 375 of MIT Media Lab., 1995.
[25] A.D. Wilson and A.F. Bobick, “Recognition and interpretation of parametric gesture,” Proc. Int. Conf. Computer Vision, 1998.

Appendix A: M step in EM learning

The parameters maximizing L can be obtained as the followings:

$$\begin{aligned}
\Phi_{i,j} &= \frac{\sum_{t=2}^{T} p(m_{t-1}=i, m_t=j|O_T)}{\sum_{t=2}^{T} p(m_{t-1}=i|O_T)} \\
\pi_j &= p(m_1=j|O_T) \\
F_j &= (R^{(j)}R_{01}^{(j)} - R_0^{(j)}R_1^{'(j)})(R^{(j)}R_{11}^{(j)} - R_1^{(j)}R_1^{'(j)})^{-1} \\
D_j &= \frac{1}{R^{(j)}}\{R_0^{(j)} - (R^{(j)}R_{01}^{(j)} - R_0^{(j)}R_1^{'(j)}) \\
&\quad \cdot(R^{(j)}R_{11}^{(j)} - R_1^{(j)}R_1^{'(j)})^{-1}R_1^{(j)}\} \\
Q_j &= \frac{\sum_{t=2}^{T}\sum_{i=1}^{M} p(m_{t-1}=i, m_t=j|O_T)\eta_{t|T}^{(i,j)}\eta_{t|T}^{'(i,j)}}{\sum_{t=2}^{T}\sum_{i=1}^{M} p(m_{t-1}=i, m_t=j|O_T)}
\end{aligned}$$

where

$$\begin{aligned}
R_{01}^{(j)} &= \sum_{t=2}^{T}\sum_{i=1}^{M} p(m_{t-1}=i, m_t=j|O_T)x_{t|T}^{(i,j)}x_{t-1|T}^{'(i)} \\
R_{11}^{(j)} &= \sum_{t=2}^{T}\sum_{i=1}^{M} p(m_{t-1}=i, m_t=j|O_T)x_{t-1|T}^{(i)}x_{t-1|T}^{'(i)} \\
R_0^{(j)} &= \sum_{t=2}^{T}\sum_{i=1}^{M} p(m_{t-1}=i, m_t=j|O_T)x_{t|T}^{(i,j)} \\
R_1^{(j)} &= \sum_{t=2}^{T}\sum_{i=1}^{M} p(m_{t-1}=i, m_t=j|O_T)x_{t-1|T}^{(i)} \\
R^{(j)} &= \sum_{t=2}^{T}\sum_{i=1}^{M} p(m_{t-1}=i, m_t=j|O_T)
\end{aligned}$$

Appendix B

$$\begin{aligned}
&p(o_t|m_{t-1}, m_t, O_{t-1}) \\
&\quad = \int_{x_t} p(o_t|x_t)p(x_t|m_{t-1}, m_t, O_{t-1})dx_t \\
&\quad = E_{p(x_t|m_{t-1},m_t,O_{t-1})}[p(o_t|x_t)] \qquad \text{(A· 1)}
\end{aligned}$$

From (A· 1) the probability of the filtered joint-discrete state is derived by

$$\begin{aligned}
&p(m_{t-1}, m_t|O_t) \\
&\quad = \frac{p(m_{t-1}, m_t|O_{t-1})E_{p(x_t|m_{t-1},m_t,O_{t-1})}[p(o_t|x_t)]}{p(o_t|O_{t-1})}. \qquad \text{(A· 2)}
\end{aligned}$$

From (3) since we assumed that

$$p(x_t|m_{t-1}, m_t, O_{t-1}) = N(x_{t|t-1}^{(m_{t-1},m_t)}, P_{t|t-1}^{(m_{t-1},m_t)}),$$

it follows that

$$\begin{aligned}
&E_{p(x_t|m_{t-1},m_t,O_{t-1})}[p(o_t|x_t)] \\
&\qquad \simeq p(o_t|x_{t|t-1}^{(m_{t-1},m_t)}). \qquad \text{(A· 3)}
\end{aligned}$$

Then the probability of the filtered joint-discrete state is given as

$$\begin{aligned}
&p(m_{t-1}, m_t|O_t) \\
&\quad \simeq k_t p(m_{t-1}, m_t|O_{t-1})p(o_t|x_{t|t-1}^{(m_{t-1},m_t)}). \qquad \text{(A· 4)}
\end{aligned}$$

Appendix C

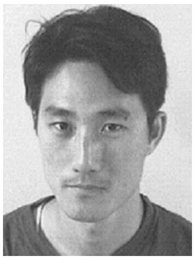
From the following (A· 5) – (A· 8), (19) can be obtained.

$$\begin{aligned}\log p(M_T, X_T, O_T|\lambda) \\ = \sum_{j=1}^{M} \psi_1(j)\{\log \pi_j + \log p_j(x_1) + \log p_j(o_1|x_1)\} \\ + \sum_{t=2}^{T}\sum_{j=1}^{M} \psi_t(j) \log p_j(o_t|x_t) \\ + \sum_{t=2}^{T}\sum_{i,j=1}^{M} \psi_{t-1}(i)\psi_t(j)\{\log \Phi_{i,j} + \log p_{i,j}(x_t|x_{t-1})\}\end{aligned} \tag{A· 5}$$

$$\begin{aligned}E_{\bar{p}}[\log p_{i,j}(x_t|x_{t-1})] \\ \simeq \frac{1}{2}\{det(Q_j^{-1}) - \eta'^{(i,j)}_{t|T} Q_j^{-1} \eta^{(i,j)}_{t|T}\} - d \log \sqrt{2\pi}\end{aligned} \tag{A· 6}$$

$$E_{\bar{p}}[\psi_1(j)] \simeq p(m_1 = j|O_T) \tag{A· 7}$$

$$E_{\bar{p}}[\psi_{t-1}(i)\psi_t(j)] \simeq p(m_{t-1} = i, m_t = j|O_T) \tag{A· 8}$$



Mun-Ho Jeong received the B.E. and M.E. degrees from KAIST in 1994 and 1996, respectively, and received the Ph.D. degree from Osaka University in 2002. Since 2002 he has been worked at Toyota Motor Corporation.

Yoshinori Kuno received the B.S., M.S. and Ph.D. degrees in 1977, 1979 and 1982, respectively, all in electrical and electronics engineering from Tokyo University. In 1982 he joined Toshiba Corporation. From 1987 to 1988, he was a Visiting Research Scientist at Carnegie Mellon University. Since 1993, he had been an Associate Professor in the Department of Mechanical Engineering for Computer-Controlled Machinery, Osaka University. Since 2001 he has been a Professor in the Department of Information and Computer Sciences, Saitama University. He is a member of Inf. Proc. Soc. Jap., Robotics Soc. Jap., Soc. Artif. Intel. And IEEE.

Nobutaka Shimada received the B.E., M.E. and Ph.D. degrees from Osaka University in 1992, 1994 and 1997, respectively. Since 1997 he has been a research associate of Department of Mechanical Engineering for Computer-Controlled Machinery, Osaka University. He is a member of IPSJ, and IEEE (Computer Society).

Yoshiaki Shirai received the B.E. degree from Nagoya University in 1964, and received the M.E. and the Ph.D. degree from the Tokyo University in 1966 and 1969, respectively. In 1969 he joined the Electrotechnical Laboratory. From 1988 he has been a Professor of the Department of Mechanical Engineering for Computer-Controlled Machinery, Osaka University. His research area has been computer vision, robotics and artificial intelligence. He is a member of the IEEE Computer Society, Information Processing Society of Japan, the JSME, the Japanese Society of Robotics, SICE, and Japanese Society of Artificial Intelligence.