

## PAPER

# Image Recognition Based on Separable Lattice Trajectory 2-D HMMs

Akira TAMAMORI<sup>†a)</sup>, Nonmember, Yoshihiko NANKAKU<sup>†b)</sup>, and Keiichi TOKUDA<sup>†c)</sup>, Members

**SUMMARY** In this paper, a novel statistical model based on 2-D HMMs for image recognition is proposed. Recently, separable lattice 2-D HMMs (SL2D-HMMs) were proposed to model invariance to size and location deformation. However, their modeling accuracy is still insufficient because of the following two assumptions, which are inherited from 1-D HMMs: i) the stationary statistics within each state and ii) the conditional independent assumption of state output probabilities. To overcome these shortcomings in 1-D HMMs, trajectory HMMs were proposed and successfully applied to speech recognition and speech synthesis. This paper derives 2-D trajectory HMMs by reformulating the likelihood of SL2D-HMMs through the imposition of explicit relationships between static and dynamic features. The proposed model can efficiently capture dependencies between adjacent observations without increasing the number of model parameters. The effectiveness of the proposed model was evaluated in face recognition experiments on the XM2VTS database.

**key words:** image recognition, hidden Markov models, separable lattice 2-D HMMs, trajectory HMMs

## 1. Introduction

With the wide spread of computers in recent years, the development of a human interface that utilize visual and auditory information is expected. It can be used to communicate with others in the same way as humans. In particular, speech recognition and image recognition are important basic technologies for this interface and research has been conducted actively. Moreover, with the recent advances of computer hardware and information technology, statistical approaches based on huge amounts of data are becoming the mainstream in many research fields. For speech recognition, Hidden Markov model (HMM) based techniques have been established [1]. However, in the field of image recognition, various approaches have been mushrooming due to the variety of the recognition objects and the complexity of data. Therefore, it is valuable to construct the general statistical models for image recognition similar to HMMs for speech recognition, which can be applied to various tasks such as face recognition, hand-written character recognition, gesture recognition, and lip reading.

The previous research of image recognition can be roughly classified into the following two: i) techniques developed by utilizing task-dependent information and ii)

techniques considering image recognition as pattern classification problems on multi-dimensional feature space objectively. The former techniques take account of the practicality and high recognition performance can be obtained even if a small amount of training data is available. On the other hand, the latter techniques should be selected when considering the general framework of image recognition. However, the pre-processings such as segmentation, normalization and feature extraction are still required to deal with the image recognition problem as pattern classification problem. These pre-processings have not been considered in many studies on the latter techniques and the heuristic normalization techniques have been applied. Additionally, the final objective in image recognition is not to accurately normalize images for human perception but to achieve better recognition performance. Therefore, it is a good idea to integrate the normalization processes into classifiers and optimize them based on a consistent criterion to improve recognition performance.

HMM based techniques for image recognition have been proposed to reduce the influence of geometric variations [2]–[12]. Geometric matching between input images and model parameters is represented by discrete hidden variables, and the normalization process is included in calculating probabilities. For an earlier work, Samaria et al. applied HMMs to human face identification tasks [2]. The observation sequence was composed of over-lapping window/line blocks extracted from each sample image and modeled by ergodic/top-to-bottom HMMs, provided that image data had to be treated as if it was 1-D data sequence. This leads to lack of robustness to geometric variations. It was therefore natural for many researchers to consider extending HMMs to multi-dimensional ones.

However, the above extension generally leads to an exponential increase in the amount of computation for its training algorithm. To reduce the computational complexity, the model structure needs to be constrained by limiting the number of possible alignments and assuming independence between hidden variables. For such model structures, pseudo 2-D HMMs [3] (embedded HMMs [4]) were proposed and applied to many image recognition tasks. A pseudo 2-D HMM has a composite state structure for a better 2-D representation while avoiding the complexity burden of a fully connected 2-D HMM. The states of a superior HMM in the horizontal direction are called super-states and each super-state has a one-dimensional HMM in the vertical direction instead of a probability density function. This assumption

Manuscript received October 29, 2013.

Manuscript revised February 24, 2014.

<sup>†</sup>The authors are with the Department of Computer Science, Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.

a) E-mail: mataki@sp.nitech.ac.jp

b) E-mail: nankaku@sp.nitech.ac.jp

c) E-mail: tokuda@sp.nitech.ac.jp

DOI: 10.1587/transinf.E97.D.1842

reduces the computational complexity and the maximum likelihood training algorithm has been proposed [5]. However, the state alignments of consecutive observation lines in the vertical direction are calculated independently of each other and this assumption does not always hold true in practice.

Essentially, the studies of 2-D dynamic programming (2D-DP) treat the same problem of the 2-D HMMs. The main difference between these studies is the definition of the cost function; The 2D-DP focuses on finding the mapping between two images with a pre-defined cost function, while the likelihood of 2-D HMMs is defined between an input image and the distribution which is estimated from multiple training images. Although some efficient approximation algorithms have been proposed for the 2D-DP problem [13]–[16], they still need high complicated costs and prior knowledge to determine the cost function is required for representing an accurate elastic matching dependently on image variations.

For another HMM based approach, separable lattice 2-D HMMs (SL2D-HMMs) were proposed [8] to reduce computational complexity while retaining good properties that model multi-dimensional data. Furthermore, hidden Markov eigenface models have been proposed [9] where the eigenface methods are integrated into SL2D-HMMs. SL2D-HMMs can perform elastic matching both horizontally and vertically, which makes it possible to model not only invariance to the size and location of an object but also nonlinear warping in all dimensions. Nevertheless, due to the model structure of SL2D-HMMs which consists of two independent 1-D Markov chains, SL2D-HMMs have the same constraints as 1-D HMMs [17] in that (i) the statistics of each state do not change dynamically and (ii) the output probability of an observation vector depends only on the current state, not on any other states nor observations.

To overcome the above shortcomings, it has been confirmed that augmenting the dimensionality of an acoustic static feature vector (e.g., cepstral coefficients) by appending its dynamic feature vectors (e.g., 1st and 2nd order delta cepstral coefficients) [18] can enhance the performance of HMM-based speech recognizers. It can be considered that augmented feature vectors can capture dependencies between adjacent acoustic feature vectors. Based on this knowledge, SL2D-HMMs can also enhance the recognition performance by appending dynamic features [12], [19], where first-order derivative coefficients in horizontal and vertical direction were applied. However, static and dynamic features are assumed to be independent variables and the relationships between them are ignored even though these relationships are essentially deterministic. As a result, inconsistency between the static and dynamic features is tolerated.

In previous work [20], trajectory HMMs were proposed and successfully applied to speech recognition and speech synthesis. The standard HMM is reformulated by imposing the explicit relationship between static and dynamic features, in order that the constraint of HMMs such as the

conditional independence and the constant statistics in each state can be relaxed. In this paper, we propose a novel generative model that reformulates SL2D-HMMs as a trajectory model, referred to as separable lattice trajectory 2-D HMMs (SLT2D-HMMs). The proposed model can overcome the shortcomings of SL2D-HMMs and capture the dependencies of adjacent observations, without increasing the number of model parameters. Consequently, the modeling ability can be significantly improved.

The rest of the paper is organized as follows. In Sect. 2, SL2D-HMMs are explained briefly. In Sect. 3, the structure of the proposed model is defined. In Sect. 4, we derive the training algorithm for the proposed model. In Sect. 5, we describe face recognition experiments on the XM2VTS database [21] and finally conclude in Sect. 6.

## 2. Separable Lattice 2-D HMMs

Separable lattice 2-D hidden Markov models (SL2D-HMMs) [8] are defined for modeling two-dimensional data. The observations of two-dimensional data, e.g., pixel values of an image are assumed to be given on a two-dimensional lattice:

$$\mathbf{O} = \{\mathbf{O}_t \mid \mathbf{t} = (t^{(1)}, t^{(2)}) \in \mathbf{T}\}, \quad (1)$$

where  $\mathbf{t}$  denotes the coordinates of the lattice in two dimensional space  $\mathbf{T}$  and  $t^{(m)} = 1, \dots, T^{(m)}$  is the coordinate of the  $m$ -th dimension. The observation  $\mathbf{O}_t$  is emitted from the state indicated by the hidden variable  $\mathbf{S}_t \in \mathbf{K}$ . The hidden variables  $\mathbf{S}_t \in \mathbf{K}$  can take one of  $K = K^{(1)}K^{(2)}$  states, which are assumed to be arranged on a two-dimensional state lattice  $\mathbf{K} = \{(1, 1), (1, 2), \dots, (1, K^{(2)}), (2, 1), \dots, (K^{(1)}, K^{(2)})\}$ . In other words, a set of hidden variables,  $\{\mathbf{S}_t \mid \mathbf{t} \in \mathbf{T}\}$ , represents a segmentation of observations into the  $K$  states, and each state corresponds to a segmented region in which the observation vectors are assumed to be generated from the same distribution. Since the observation  $\mathbf{O}_t$  is dependent only on the state  $\mathbf{S}_t$  as in ordinary HMMs, dependencies among hidden variables determine the properties and modeling ability of two-dimensional HMMs.

To reduce the number of possible state sequences, the hidden variables of a SL2D-HMM are constrained to be composed of two Markov chains:

$$\begin{aligned} \mathbf{S} &= \{\mathbf{S}^{(1)}, \mathbf{S}^{(2)}\}, \\ \mathbf{S}^{(m)} &= \{S_1^{(m)}, \dots, S_{t^{(m)}}^{(m)}, \dots, S_{T^{(m)}}^{(m)}\}, \end{aligned} \quad (2) \quad (3)$$

where  $\mathbf{S}^{(m)}$  is the Markov chain along with the  $m$ -th coordinate and  $S_{t^{(m)}}^{(m)} \in \{1, \dots, K^{(m)}\}$ . In the separable lattice 2-D HMMs, the composite structure of hidden variables is defined as the product of hidden state sequences:  $\mathbf{S}_t = (S_{t^{(1)}}^{(1)}, S_{t^{(2)}}^{(2)})$ . This means that the segmented regions of observations are constrained to be rectangles and this allows an observation lattice to be elastic in both vertical and horizontal directions. Using this structure, the number of possible state sequences can be reduced from  $\{\prod_m K^{(m)}\}^{\prod_m T^{(m)}}$  to  $\prod_m \{K^{(m)}\}^{T^{(m)}}$ . Figures 1 and 2 show the model structure and graphical model representation of SL2D-HMMs,

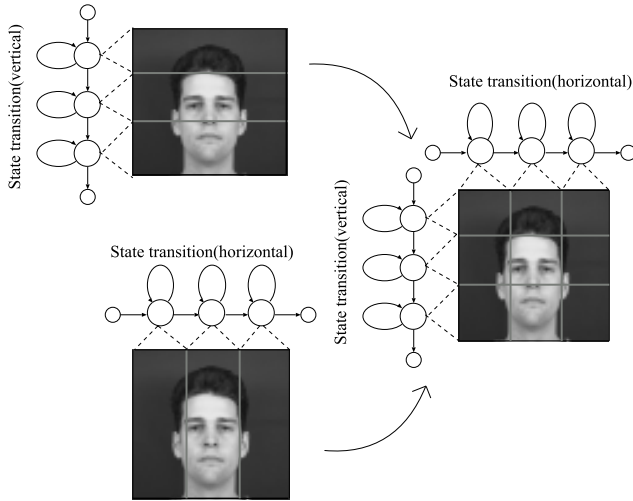


Fig. 1 Model structure of the separable lattice 2-D HMMs: hidden state sequences are composed of independent two Markov chains.

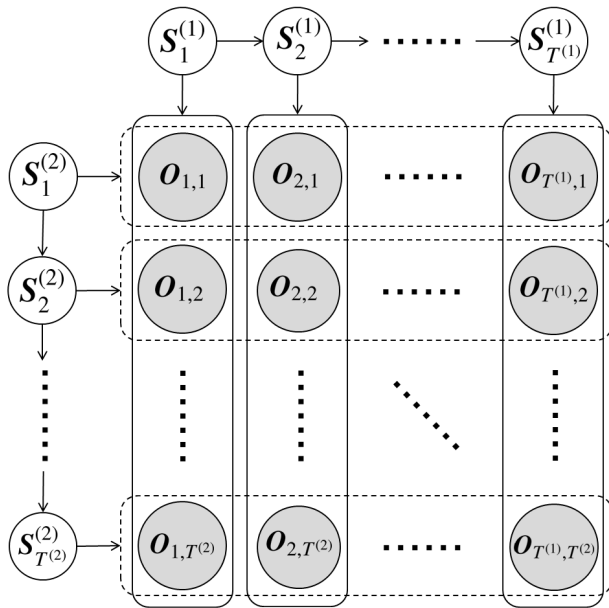


Fig. 2 Graphical model representation of the separable lattice 2-D HMMs: The rounded box represents a group of variables and the arrow to the box represents the dependency to all variables in the box instead of drawing arrows to the all variables. The observations are emitted from the product of horizontal and vertical hidden state sequences.

respectively.

In SL2D-HMMs, the joint probability of observation vectors  $\mathbf{O}$  and hidden variables  $\mathbf{S}$  can be written as

$$\begin{aligned}
 P(\mathbf{O}, \mathbf{S} | \Lambda) & \\
 &= P(\mathbf{O} | \mathbf{S}, \Lambda) \cdot \prod_{m=1,2} P(\mathbf{S}^{(m)} | \Lambda) \\
 &= \prod_t P(\mathbf{O}_t | \mathbf{S}_t, \Lambda) \\
 &\quad \times \prod_{m=1,2} \left[ P(S_1^{(m)} | \Lambda) \prod_{t^{(m)}=2}^{T^{(m)}} P(S_{t^{(m)}}^{(m)} | S_{t^{(m)}-1}^{(m)}, \Lambda) \right] \quad (5)
 \end{aligned}$$

where  $\Lambda$  is a set of model parameters of SL2D-HMMs. The model parameters of SL2D-HMMs are summarized as follows:

- **Parameters for state transition probability:**

- 1)  $\Pi^{(m)} = \{\pi_i^{(m)} | 1 \leq i \leq K^{(m)}\}$ : the initial state probability distribution, where

$$\pi_i^{(m)} = P(S_1^{(m)} = i | \Lambda) \quad (6)$$

is the probability of state  $i$  at  $t^{(m)} = 1$  in the  $m$ -th state sequence  $\mathbf{S}^{(m)}$ .

- 2)  $\mathbf{A}^{(m)} = \{a_{ij}^{(m)} | 1 \leq i, j \leq K^{(m)}\}$ : the transition probability matrix, where

$$a_{ij}^{(m)} = P(S_{t^{(m)}}^{(m)} = j | S_{t^{(m)}-1}^{(m)} = i, \Lambda) \quad (7)$$

is the transition probability from state  $i$  to state  $j$  in the  $m$ -th state sequence  $\mathbf{S}^{(m)}$ .

- **Parameters for output probability distribution:**

$\mathbf{B} = \{b_k(\mathbf{O}_t) | k \in \mathbf{K}\}$ : the output probability distributions, where  $b_k(\mathbf{O}_t)$  is the probability of observation vector  $\mathbf{O}_t$  at the state  $k$  on the state lattice  $\mathbf{K}$  and assumed to be a single Gaussian distribution:

$$P(\mathbf{O}_t | S_t = k) = \mathcal{N}(\mathbf{O}_t | \mu_k, \Sigma_k) \quad (8)$$

where  $\mu_k$  and  $\Sigma_k$  denote the “state level” mean vector and the covariance matrix, respectively. Note that SL2D-HMMs have  $K^{(1)}K^{(2)}$  Gaussians directly as the parameters on a lattice unlike factorial HMMs [22]<sup>†</sup>.

The evaluation of the exact likelihood is computationally intractable because the joint probability  $P(\mathbf{O}, \mathbf{S} | \Lambda)$  must be evaluated over all the possible state transitions and the order becomes  $O(\prod_m \{K^{(m)}\}^{T^{(m)}})$ . Similarly, the exact EM algorithm also becomes infeasible. To cope with this problem, the training algorithm for SL2D-HMMs using the variational EM algorithm were derived in [8], where the log-likelihood can be approximated by the variational lower bound. Although some extensions of SL2D-HMMs have been proposed, e.g., a structure for rotational variations [10], explicit state duration modeling [11], and a structure with multiple horizontal/vertical Markov chains [12], this paper uses an original form of SL2D-HMMs.

### 3. Separable Lattice Trajectory 2-D HMMs

In the previous section, we described the structure of SL2D-HMMs, where the hidden variables are composed of two independent 1-D Markov chains. Therefore, similar to the 1-D HMMs, the following two limitations are imposed on SL2D-HMMs [17]:

- i) The statistics of each state do not change dynamically.

<sup>†</sup>Factorial HMMs have  $(K^{(1)} + K^{(2)})$  Gaussians along with Markov chains and they contribute linearly to the output probability distributions.

- ii) The output probability of the observation is conditionally independent, given the horizontal and vertical states.

To overcome these shortcomings, augmenting the dimensionality of static feature vectors (e.g., pixel values) by appending their dynamic feature vectors (e.g., delta and delta-delta coefficients) [18] to capture dependencies between adjacent observations can enhance the performance of the HMM-based speech recognizers [23]. Generally, dynamic features are calculated as regression coefficients from their neighboring static features and can be represented as a linear combination of static features. In other words, the relationship between static and dynamic features is linear, and therefore, *deterministic*. However, this relationship is ignored and static and dynamic features are modeled as independent statistical variables in the standard HMM framework. Before deriving the proposed model, applications of dynamic feature in 1-D and 2-D case will be described in the next section. Then, in Sect. 3.2, the proposed model will be derived in order to avoid the above problem.

### 3.1 Applications of Dynamic Features

#### 3.1.1 Dynamic Features for Speech Data

This section describes dynamic features for acoustic features (e.g., Mel-Frequency Cepstral Coefficients) which were developed in 1-D time-domain. This have often been used to model speech signals by HMMs. Let  $\mathbf{o} = [\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_T]$  be the sequence of speech parameter vectors, where  $\mathbf{o}_t$  is a speech parameter vector at time  $t$ . In a typical speech recognition system, it is assumed that the speech parameter vector  $\mathbf{o}_t$  is a  $3M \times 1$  vector consisting of an  $M$ -dimensional acoustic static feature

$$\mathbf{c}_t = [c_t(1), c_t(2), \dots, c_t(M)] \quad (9)$$

and its first and second order dynamic feature vectors,  $\Delta \mathbf{c}_t$  and  $\Delta^2 \mathbf{c}_t$ , that is

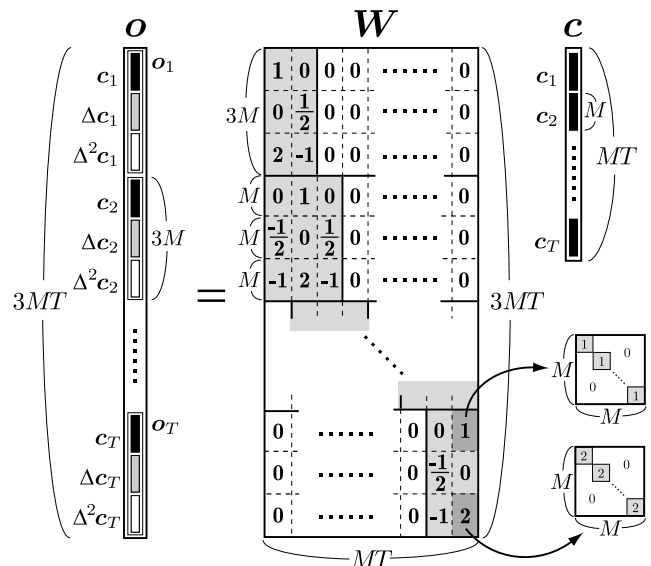
$$\mathbf{o}_t = [\mathbf{c}_t^\top, \Delta \mathbf{c}_t^\top, \Delta^2 \mathbf{c}_t^\top]^\top. \quad (10)$$

The dynamic features are often calculated as regression coefficients from their neighboring static features, i.e.,

$$\Delta \mathbf{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau) \mathbf{c}_{t+\tau}, \quad (11)$$

$$\Delta^2 \mathbf{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau) \mathbf{c}_{t+\tau}, \quad (12)$$

where  $\{w^{(d)}(\tau)\}_{\tau=-L_-^{(d)}, \dots, L_+^{(d)}}$  are window coefficients to calculate the  $d$ -th order dynamic feature. Usually, the maximum window length  $L$  is set to 1–4. The relationship between the observation vector sequence  $\mathbf{o} = [\mathbf{o}_1^\top, \mathbf{o}_2^\top, \dots, \mathbf{o}_T^\top]^\top$  and static feature sequence  $\mathbf{c} = [\mathbf{c}_1^\top, \mathbf{c}_2^\top, \dots, \mathbf{c}_T^\top]^\top$  can be arranged



**Fig. 3** An example of relationship between the observation vector sequence  $\mathbf{o}$  and the static feature vector sequence  $\mathbf{c}$  in a matrix form [20], where the dynamic feature vectors are calculated using Eqs. (11) and (12) with  $L_-^{(1)} = L_+^{(1)} = L_-^{(2)} = L_+^{(2)} = 1$ ,  $w^{(1)}(-1) = -0.5$ ,  $w^{(1)}(0) = 0.0$ ,  $w^{(1)}(1) = 0.5$ ,  $w^{(2)}(-1) = 1.0$ ,  $w^{(2)}(0) = -2.0$ ,  $w^{(2)}(1) = 1.0$ .

in a matrix form as

$$\mathbf{o} = \mathbf{W}\mathbf{c}, \quad (13)$$

where  $\mathbf{W}$  is a  $3MT \times MT$  window matrix and the elements of  $\mathbf{W}$  are given as follows:

$$\mathbf{W} = [\mathbf{W}_1 \ \dots \ \mathbf{W}_t \ \dots \ \mathbf{W}_T]^\top \otimes \mathbf{I}_{M \times M}, \quad (14)$$

$$\mathbf{W}_t = [\mathbf{w}_t^{(0)}, \mathbf{w}_t^{(1)}, \mathbf{w}_t^{(2)}], \quad (15)$$

$$\mathbf{w}_t^{(d)} = \left[ \underbrace{0, \dots, 0}_{t-L_-^{(d)}-1}, w^{(d)}(-L_-^{(d)}), \dots, w^{(d)}(0), \dots, w^{(d)}(L_+^{(d)}), \underbrace{0, \dots, 0}_{T-(t+L_+^{(d)})} \right]^\top, \quad d = 0, 1, 2 \quad (16)$$

where  $L_-^{(0)} = L_+^{(0)} = 0$ ,  $\mathbf{w}^{(0)} = 1$ , and  $\otimes$  denotes the Kronecker product for matrices. An example of the relationship is shown in Fig. 3.

#### 3.1.2 Dynamic Features for Image Data

In 2-D image case, the observation vector  $\mathbf{O}_t$  is assumed to consist of the  $M$ -dimensional static feature vector

$$\mathbf{C}_t = [C_t(1), C_t(2), \dots, C_t(M)]^\top \quad (17)$$

and horizontal/vertical dynamic feature vectors,  $\Delta^{(H)} \mathbf{C}_t$  and  $\Delta^{(V)} \mathbf{C}_t$ , that is<sup>†</sup>

$$\mathbf{O}_t = [\mathbf{C}_t^\top, \Delta^{(H)} \mathbf{C}_t^\top, \Delta^{(V)} \mathbf{C}_t^\top]^\top, \quad (18)$$

<sup>†</sup>Using higher-order dynamic features is straightforward. Moreover, dynamic features in other directions, e.g., diagonal dynamic features can be adopted easily.

where  $\mathbf{t} = (t^{(1)}, t^{(2)})$ . Likewise 1-D case described in the previous section, these dynamic features are calculated as regression coefficients from their neighboring static features:

$$\Delta^{(H)}\mathbf{C}_t = \sum_{\tau=-L_-^{(H)}}^{L_+^{(H)}} w^{(H)}(\tau)\mathbf{C}_{(t^{(1)}+\tau, t^{(2)})}, \quad (19)$$

$$\Delta^{(V)}\mathbf{C}_t = \sum_{\tau=-L_-^{(V)}}^{L_+^{(V)}} w^{(V)}(\tau)\mathbf{C}_{(t^{(1)}, t^{(2)}+\tau)}, \quad (20)$$

where  $\{w^{(H)}(\tau)\}_{\tau=-L_-^{(H)}, \dots, L_+^{(H)}}$  and  $\{w^{(V)}(\tau)\}_{\tau=-L_-^{(V)}, \dots, L_+^{(V)}}$  are window coefficients to calculate the horizontal and vertical dynamic features, respectively. The observation vectors and static feature vectors on the 2-D lattice can be rewritten in  $MT^{(1)}T^{(2)}$  size vector forms as

$$\mathbf{O} = \begin{bmatrix} \mathbf{O}_{(1,1)}^\top & \dots & \mathbf{O}_t^\top & \dots & \mathbf{O}_{(T^{(1)}, T^{(2)})}^\top \end{bmatrix}^\top, \quad (21)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{(1,1)}^\top & \dots & \mathbf{C}_t^\top & \dots & \mathbf{C}_{(T^{(1)}, T^{(2)})}^\top \end{bmatrix}^\top, \quad (22)$$

where both elements of  $\mathbf{O}$  and  $\mathbf{C}$  are aligned in raster order of the 2-D lattice.

A linear relationship between  $\mathbf{O}$  and  $\mathbf{C}$  in 2-D case, which is similar to Eq. (13) in 1-D case, can be obtained as

$$\mathbf{O} = \mathbf{W}\mathbf{C}, \quad (23)$$

where  $\mathbf{W}$  is a  $3MT^{(1)}T^{(2)} \times MT^{(1)}T^{(2)}$  window matrix given as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{(1,1)} & \dots & \mathbf{W}_t & \dots & \mathbf{W}_{(T^{(1)}, T^{(2)})} \end{bmatrix}^\top \otimes \mathbf{I}_{M \times M}, \quad (24)$$

$$\mathbf{W}_t = [\mathbf{w}_t^{(S)}, \mathbf{w}_t^{(H)}, \mathbf{w}_t^{(V)}], \quad (25)$$

where  $\mathbf{w}_t^{(S)}$ ,  $\mathbf{w}_t^{(H)}$ , and  $\mathbf{w}_t^{(V)}$  are  $T^{(1)}T^{(2)}$  size vectors. They are defined so that following relationships are satisfied based on Eqs. (18), (19), (20) and (25):

$$\mathbf{C}_t = (\mathbf{w}_t^{(S)})^\top \otimes \mathbf{I}_{M \times M} \mathbf{C}, \quad (26)$$

$$\Delta \mathbf{C}_t^{(H)} = (\mathbf{w}_t^{(H)})^\top \otimes \mathbf{I}_{M \times M} \mathbf{C}, \quad (27)$$

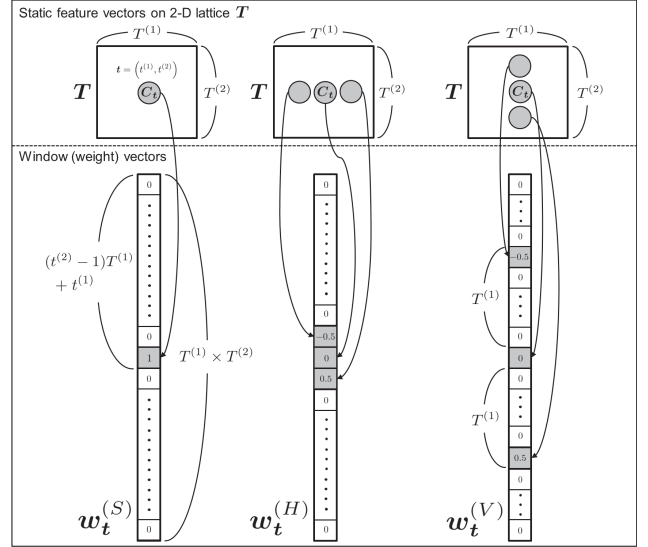
$$\Delta \mathbf{C}_t^{(V)} = (\mathbf{w}_t^{(V)})^\top \otimes \mathbf{I}_{M \times M} \mathbf{C}, \quad (28)$$

$$\mathbf{O}_t = (\mathbf{W}_t^\top \otimes \mathbf{I}_{M \times M}) \mathbf{C}. \quad (29)$$

The functions of window vectors  $\mathbf{w}_t^{(S)}$ ,  $\mathbf{w}_t^{(H)}$ , and  $\mathbf{w}_t^{(V)}$  can be explained as follows: From Eq. (26),  $\mathbf{w}_t^{(S)}$  is a vector which extract the static feature vector at  $\mathbf{t} = (t^{(1)}, t^{(2)})$  from image data. Furthermore, from Eqs. (27) and (28),  $\mathbf{w}_t^{(H)}$  and  $\mathbf{w}_t^{(V)}$  are vectors which extract the gradients of horizontal and vertical direction centered at  $\mathbf{t}$ , respectively. Examples of  $\mathbf{w}_t^{(S)}$ ,  $\mathbf{w}_t^{(H)}$ , and  $\mathbf{w}_t^{(V)}$  are shown in Fig. 4, where the maximum window length  $L = 1$  and  $M = 1$  for simplicity.

### 3.2 Model Definition

Based on the relationship  $\mathbf{O}$  and  $\mathbf{C}$  in Eq. (23), the definition



**Fig. 4** Examples of  $\mathbf{w}_t^{(S)}$ ,  $\mathbf{w}_t^{(H)}$ , and  $\mathbf{w}_t^{(V)}$ , where  $L_-^{(H)} = L_+^{(H)} = L_-^{(V)} = L_+^{(V)} = 1$ ,  $w^{(H)}(-1) = w^{(V)}(-1) = -0.5$ ,  $w^{(H)}(0) = w^{(V)}(0) = 0.0$ ,  $w^{(H)}(1) = w^{(V)}(1) = 0.5$  from Eqs. (19) and (20). The circles in the top box represent the static features. Also, the squares in the bottom box represent the elements of each window vector. The arrow from the top to the bottom represents a multiplication between the corresponding static feature vector and the element of window vector. The resultants of those sums are dynamic feature vectors as shown in Eqs. (26), (27), and (28).

of the proposed model can be derived. The output probability  $P(\mathbf{O} | \mathbf{S}, \Lambda)$  of SL2D-HMMs is given by

$$P(\mathbf{O} | \mathbf{S}, \Lambda) = \mathcal{N}(\mathbf{O} | \boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S) = \prod_t \mathcal{N}(\mathbf{O}_t | \boldsymbol{\mu}_{S_t}, \boldsymbol{\Sigma}_{S_t}), \quad (30)$$

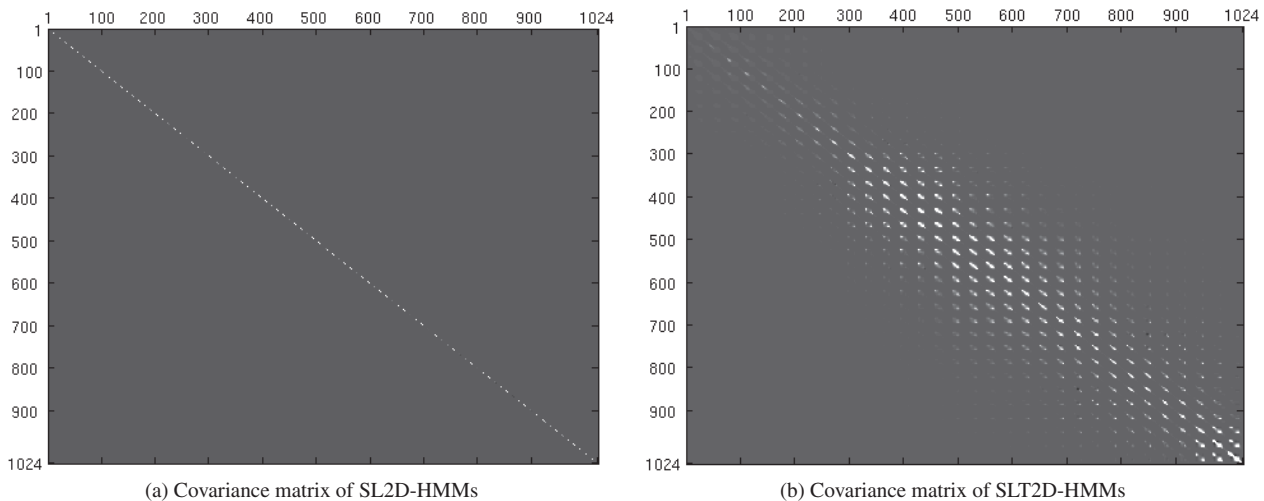
where  $\mathcal{N}(\cdot | \boldsymbol{\mu}, \boldsymbol{\Sigma})$  denotes the Gaussian distribution with a mean vector  $\boldsymbol{\mu}$  and a covariance matrix  $\boldsymbol{\Sigma}$ , and  $\boldsymbol{\mu}_S$  and  $\boldsymbol{\Sigma}_S$  are the “image level” mean vector and covariance matrix given state sequences  $\mathbf{S}$ , respectively. They are constructed by concatenating the “state level” mean vectors and covariance matrices in accordance with state sequences  $\mathbf{S}$ :

$$\boldsymbol{\mu}_S = \begin{bmatrix} \boldsymbol{\mu}_{S_{(1,1)}}^\top & \dots & \boldsymbol{\mu}_{S_t}^\top & \dots & \boldsymbol{\mu}_{S_{(T^{(1)}, T^{(2)})}}^\top \end{bmatrix}^\top, \quad (31)$$

$$\boldsymbol{\Sigma}_S = \begin{bmatrix} \boldsymbol{\Sigma}_{S_{(1,1)}} & & & & \mathbf{0} \\ & \ddots & & & \\ & & \boldsymbol{\Sigma}_{S_t} & & \\ & & & \ddots & \\ \mathbf{0} & & & & \boldsymbol{\Sigma}_{S_{(T^{(1)}, T^{(2)})}} \end{bmatrix}. \quad (32)$$

However, Eq. (30) becomes an invalid probabilistic distribution over  $\mathbf{C}$  because the integral of Eq. (30) over  $\mathbf{C}$  is not equal to 1. Namely, Eq. (30) is not normalized as the probability distribution of  $\mathbf{C}$ . To yield a valid probability distribution over  $\mathbf{C}$ , Eq. (30) can be re-normalized and written as

$$P(\mathbf{C} | \mathbf{S}, \Lambda) = \frac{1}{Z_S} \mathcal{N}(\mathbf{W}\mathbf{C} | \boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S) = \mathcal{N}(\mathbf{C} | \bar{\mathbf{C}}_S, \mathbf{P}_S), \quad (33)$$



**Fig. 5** Examples of covariance matrix. (a) shows the covariance matrix  $\Sigma_S$  of SL2D-HMMs in Eq. (32) and (b) shows the covariance matrix  $P_S$  of SLT2D-HMMs in Eq. (36) where static, 1st order horizontal and vertical dynamic feature vectors were applied. They were estimated from pixel values of face images where the size of the face images was  $32 \times 32$ . The rows and columns are aligned in raster order of the 2-D lattice (see Fig. 2).

$$\begin{aligned}
 Z_S &= \int \mathcal{N}(\mathbf{WC} | \mu_S, \Sigma_S) d\mathbf{C} & (34) \\
 &= \frac{\sqrt{(2\pi)^{MT^{(1)}T^{(2)}} |\mathbf{P}_S|}}{\sqrt{(2\pi)^{3MT^{(1)}T^{(2)}} |\Sigma_S|}} \\
 &\quad \times \exp \left\{ -\frac{1}{2} \left( \mu_S^\top \Sigma_S^{-1} \mu_S - \mathbf{r}_S^\top \mathbf{P}_S \mathbf{r}_S \right) \right\}, & (35)
 \end{aligned}$$

where  $Z_S$  is a normalization term, and  $\bar{\mathbf{C}}_S$  and  $\mathbf{P}_S$  are the  $MT^{(1)}T^{(2)}$  mean vector and the  $MT^{(1)}T^{(2)} \times MT^{(1)}T^{(2)}$  covariance matrix, respectively. Also,  $\mathbf{r}_S$ ,  $\bar{\mathbf{C}}_S$  and  $\mathbf{P}_S$  are given as

$$\mathbf{R}_S = \mathbf{W}^\top \Sigma_S^{-1} \mathbf{W} = \mathbf{P}_S^{-1}, \quad (36)$$

$$\mathbf{r}_S = \mathbf{W}^\top \Sigma_S^{-1} \mu_S, \quad (37)$$

$$\bar{\mathbf{C}}_S = \mathbf{P}_S \mathbf{r}_S. \quad (38)$$

Using the above distribution, the joint distribution of static feature vectors  $\mathbf{C}$  and hidden variables  $\mathbf{S}$  can be written as:

$$P(\mathbf{C}, \mathbf{S} | \Lambda) = P(\mathbf{C} | \mathbf{S}, \Lambda) \prod_{m=1,2} P(\mathbf{S}^{(m)} | \Lambda). \quad (39)$$

In the proposed model, the hidden variables are composed of two independent Markov chains, similar to SL2D-HMMs (see Eq. (2)). Therefore,  $P(\mathbf{S} | \Lambda)$  can be factorized into the product of horizontal and vertical state transition probabilities, as shown in Eq. (39). By marginalizing  $P(\mathbf{C}, \mathbf{S} | \Lambda)$  over all possible state sequences  $\mathbf{S}$ , SL2D-HMMs can be reformulated as follows:

$$\begin{aligned}
 P(\mathbf{C} | \Lambda) &= \sum_{\mathbf{S}} P(\mathbf{C}, \mathbf{S} | \Lambda) \\
 &= \sum_{\mathbf{S}} P(\mathbf{C} | \mathbf{S}, \Lambda) \prod_{m=1,2} P(\mathbf{S}^{(m)} | \Lambda), & (40)
 \end{aligned}$$

$$P(\mathbf{C} | \mathbf{S}, \Lambda) = \frac{1}{Z_S} \prod_t \mathcal{N}(\mathbf{WC}_t | \mu_{S_t}, \Sigma_{S_t}) \quad (41)$$

$$= \frac{1}{Z_S} \mathcal{N}(\mathbf{WC} | \mu_S, \Sigma_S) \quad (42)$$

$$= \mathcal{N}(\mathbf{C} | \bar{\mathbf{C}}_S, \mathbf{P}_S), \quad (43)$$

where  $\Lambda$  is a set of model parameters of the proposed model. In this paper, the proposed model is referred to as separable lattice trajectory 2-D HMMs (SLT2D-HMMs). The term ‘‘trajectory’’ suggests that the above formalization of the proposed model is analogous to that of 1-D trajectory HMMs and the advantageous properties will also be inherited to the proposed model as well. It should be noted that the summation over  $\mathbf{S}$  in Eq. (40) can be performed by  $O\left(\prod_m \{K^{(m)}\}^{T^{(m)}}\right)$ , which is the exactly same order as SL2D-HMMs. Therefore, similar to SL2D-HMMs, the evaluation of the exact likelihood of the proposed model is computationally intractable. In Sect. 4, a strategy will be described to make this problem computationally tractable. It should be also noted that covariance matrix  $\mathbf{P}_S$  is generally full even when using the completely same model parameter set as SL2D-HMMs. Therefore, the inter-pixel correlation can be modeled by the covariance matrix  $\mathbf{P}_S$ . As a result, the proposed model can mitigate the limitations of SL2D-HMMs.

Figure 5 shows examples of covariance matrix  $\Sigma_S$  of SL2D-HMMs and covariance matrix  $\mathbf{P}_S$  of SLT2D-HMMs in which static, 1st order horizontal and vertical dynamic feature vectors were applied. The covariance matrix was estimated from pixel values of face images, where the size of the face images was  $32 \times 32$ . The detail of the training data and conditions will be described in Sect. 5.1. Note that both the rows and columns are aligned in raster order of the 2-D lattice (see Eq. (1) and Fig. 2), because the rows of  $\mathbf{C}$  in Eq. (22) are aligned in raster order. In both figures, white color represents higher value and black color represents lower value. It can be observed from Fig. 5(a) that only diagonal elements have higher value. On the other

hand, from Fig. 5 (b), it can be observed that not only diagonal elements but also non-diagonal, especially, band-diagonal elements have higher value. This is the one of the evidences that SLT2D-HMMs can capture the correlation of adjacent observations, while SL2D-HMMs cannot capture it.

### 3.3 Relation to Other Statistical Models

It has been discussed in [24] that there exists the relationship between the trajectory HMMs [20] and the product of experts (PoE) [25], especially, product of Gaussian experts (PoG) [26]. PoE combines multiple models by taking their product in the likelihood and normalizing it to form a new likelihood function. It can be viewed as an intersection of all distribution while MoE [27] which combines each models by summation can be viewed as a union of all models. PoG is a particular case of PoE where each expert is an unnormalized Gaussian, and Gaussian Mixture model (GMM) [28] is a particular case of MoE where each expert is a normalized Gaussian. According to [24], PoE (PoG) is an efficient way to represent high-dimensional data which simultaneously satisfies many different low-dimensional constraints. In Eq. (41),  $\mathcal{N}(\mathbf{WC}_t | \boldsymbol{\mu}_{S_t}, \boldsymbol{\Sigma}_{S_t})$  is an unnormalized Gaussian as a probability distribution of  $\mathbf{C}_t$ . The output probability of SLT2D-HMMs can be viewed as PoG where the relationship between static and dynamic features are modeled by Gaussian experts. The normalization term  $Z_S$  in Eq. (41) can be represented in a closed form as Eq. (35), without any approximation. Therefore, the output probability  $P(\mathbf{C} | \mathbf{S}, \Lambda)$  can be evaluated strictly and this helps the great simplification of model training, compared to the general case of PoE. This is an advantageous property of SLT2D-HMMs.

SLT2D-HMMs can also be viewed as hidden Gaussian Markov random fields [29] from the interesting discussion of the relationship between 1-D trajectory HMMs and Markov random fields in [24]. The graphical model representation of SLT2D-HMMs can be specified by the window matrix  $\mathbf{W}$ , where clique potential functions are given by Gaussian distributions and edges depend on cliques that are specified by the window coefficients. By changing the window matrix according to the situation, the graphical model structure of SLT2D-HMMs can be changed. This is also an advantageous property of SLT2D-HMMs.

## 4. Training Algorithm

The parameters of the proposed model can be estimated via the expectation maximization (EM) algorithm [30] which is an iterative procedure for approximating the Maximum Likelihood (ML) estimate. This algorithm maximizes the expectation of the complete data log-likelihood so called  $Q$ -function:

$$Q(\Lambda, \Lambda') = \sum_{\mathbf{S}} P(\mathbf{S} | \mathbf{C}, \Lambda) \log P(\mathbf{C}, \mathbf{S} | \Lambda'). \quad (44)$$

By maximizing the  $Q$ -function with respect to model parameters  $\Lambda$ , the re-estimation formula in the M-step can be

easily derived. However, the evaluation of the posterior distribution  $P(\mathbf{S} | \mathbf{C}, \Lambda)$  over all possible state sequences  $\mathbf{S}$  is computationally intractable due to its combination of hidden variables. In this paper, the single-path Viterbi approximation was applied to make this problem computationally tractable. As a result, the problem is broken down into the following two maximization problems:

$$\mathbf{S}_{max} = \arg \max_{\mathbf{S}} P(\mathbf{C}, \mathbf{S} | \Lambda), \quad (45)$$

$$\hat{\Lambda} = \arg \max_{\Lambda} P(\mathbf{C}, \mathbf{S}_{max} | \Lambda). \quad (46)$$

However, it is still difficult to solve the problem of Eq. (45) because the covariance matrix  $\mathbf{P}_S$  is generally full.

### 4.1 Estimation of Sub-Optimum State Sequence

In this section, the Viterbi approximation [31] to solve the maximization problem of Eq. (45) is described. This approximation is based on the following relationship

$$\mathbf{S}_{max} = \arg \max_{\mathbf{S}} P(\mathbf{C}, \mathbf{S} | \Lambda) \quad (47)$$

$$= \arg \max_{\mathbf{S}} P(\mathbf{C} | \mathbf{S}, \Lambda) P(\mathbf{S} | \Lambda) \quad (48)$$

$$= \arg \max_{\mathbf{S}} \frac{1}{Z_S} \mathcal{N}(\mathbf{O} | \boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S) P(\mathbf{S} | \Lambda) \quad (49)$$

$$\approx \arg \max_{\mathbf{S}} \mathcal{N}(\mathbf{O} | \boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S) P(\mathbf{S} | \Lambda), \quad (50)$$

where the Viterbi approximation is applied in Eq. (50). Let  $\mathbf{S}_{sub} = (\mathbf{S}_{sub}^{(1)}, \mathbf{S}_{sub}^{(2)})$  be a sub-optimum state sequence for SLT2D-HMMs. In order to obtain  $\mathbf{S}_{sub}$  from all possible state sequence, following approximation strategy was adopted in this paper:

- Step 1** Initialize  $\mathbf{S}_{sub}$  with the Viterbi state sequence  $\mathbf{S}_{vit} = (\mathbf{S}_{vit}^{(1)}, \mathbf{S}_{vit}^{(2)})$  of SL2D-HMMs.
- Step 2** Add small variations on each boundary of  $\mathbf{S}_{sub}^{(1)}$  and  $\mathbf{S}_{sub}^{(2)}$  and collect resulting state sequences as candidates. In this paper, the small variations were shift of  $\pm 1$  of bounding position.
- Step 3** Select the best state sequence from the candidates in the sense that the likelihood function is most increased.
- Step 4** Replace the current state sequence with the best state sequence.
- Step 5** If the log-likelihood function has not converged, return to **Step 2**. Otherwise, stop the iteration.

### 4.2 Estimation of Model Parameters

In this section, the maximization problem of Eq. (46) is described. The problem is equivalent to maximizing the log-likelihood

$$\begin{aligned} & \log P(\mathbf{C} | \mathbf{S}, \Lambda) \\ &= -\frac{1}{2} \left\{ \mathbf{M} \mathbf{T}^{(1)} \mathbf{T}^{(2)} \log(2\pi) - \log |\mathbf{R}_S| + \mathbf{C}^T \mathbf{R}_S \mathbf{C} \right. \\ & \quad \left. + \mathbf{r}_S^T \mathbf{P}_S \mathbf{r}_S - 2 \mathbf{r}_S^T \mathbf{C} \right\} \end{aligned} \quad (51)$$

with respect to a supervector  $\mathbf{m}$  and supermatrix  $\phi$  which are defined by concatenating the mean vectors and precision matrices of all independent states, that is

$$\mathbf{m} = \left[ \boldsymbol{\mu}_{(1,1)}^\top \quad \cdots \quad \boldsymbol{\mu}_k^\top \quad \cdots \quad \boldsymbol{\mu}_{(K^{(1)},K^{(2)})}^\top \right]^\top, \quad (52)$$

$$\phi = \left[ \boldsymbol{\Sigma}_{(1,1)}^{-1} \quad \cdots \quad \boldsymbol{\Sigma}_k^{-1} \quad \cdots \quad \boldsymbol{\Sigma}_{(K^{(1)},K^{(2)})}^{-1} \right]^\top. \quad (53)$$

We define a  $3MT^{(1)}T^{(2)} \times MK^{(1)}K^{(2)}$  matrix  $\mathbf{F}_S$  whose elements are 0 or 1 determined according to the state sequence  $S$  so that the following relationships are satisfied:

$$\boldsymbol{\mu}_S = \mathbf{F}_S \mathbf{m}, \quad \boldsymbol{\Sigma}_S^{-1} = \text{diag}[\mathbf{F}_S \phi]. \quad (54)$$

By using  $\mathbf{F}_S$ , Eqs. (36) and (37) can be written as

$$\mathbf{R}_S = \mathbf{W}^\top \cdot \text{diag}[\mathbf{F}_S \phi] \cdot \mathbf{W} = \mathbf{P}_S^{-1}, \quad (55)$$

$$\mathbf{r}_S = \mathbf{W}^\top \cdot \text{diag}[\mathbf{F}_S \phi] \cdot \mathbf{F}_S \mathbf{m}. \quad (56)$$

According to (55) and (56), Eq. (51) can be re-written as

$$\begin{aligned} & \log P(\mathbf{C} | \mathbf{S}, \Lambda) \\ &= -\frac{1}{2} \left\{ MT^{(1)}T^{(2)} \log(2\pi) - \log \left| \mathbf{W}^\top \text{diag}[\mathbf{F}_S \phi] \mathbf{W} \right| \right. \\ & \quad + \mathbf{C}^\top \mathbf{W}^\top \text{diag}[\mathbf{F}_S \phi] \mathbf{W} \mathbf{C} \\ & \quad + \mathbf{m}^\top \mathbf{F}_S^\top (\text{diag}[\mathbf{F}_S \phi]) \mathbf{W}^\top \mathbf{P}_S \mathbf{W} (\text{diag}[\mathbf{F}_S \phi]) \mathbf{F}_S \mathbf{m} \\ & \quad \left. - 2\mathbf{m}^\top \mathbf{F}_S^\top (\text{diag}[\mathbf{F}_S \phi]) \mathbf{W}^\top \mathbf{C} \right\}. \end{aligned} \quad (57)$$

Therefore, a partial derivative of Eq. (51) with respect to  $\mathbf{m}$  and  $\phi$  can be written as

$$\frac{\partial \log P(\mathbf{C} | \mathbf{S}, \Lambda)}{\partial \mathbf{m}} = \mathbf{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \mathbf{W} (\mathbf{C} - \bar{\mathbf{C}}_S), \quad (58)$$

$$\begin{aligned} \frac{\partial \log P(\mathbf{C} | \mathbf{S}, \Lambda)}{\partial \phi} &= \frac{1}{2} \mathbf{F}_S^\top \text{diag}^{-1} \left[ \mathbf{W} \mathbf{G}_S \mathbf{W}^\top \right. \\ & \quad \left. + 2\boldsymbol{\mu}_S (\mathbf{C} - \bar{\mathbf{C}}_S)^\top \mathbf{W}^\top \right], \end{aligned} \quad (59)$$

where  $\mathbf{G}_S = \mathbf{P}_S + \bar{\mathbf{C}}_S \bar{\mathbf{C}}_S^\top - \mathbf{C} \mathbf{C}^\top$  and  $\text{diag}^{-1}$  denotes the extraction of only diagonal elements from a square matrix. By setting Eq. (58) equals to  $\mathbf{0}_{3MK^{(1)}K^{(2)}}$  and solving the resultant linear equation, the following re-estimation formula for the supervector  $\mathbf{m}$  maximizing Eq. (51) can be obtained:

$$\hat{\mathbf{m}} = \mathbf{A}^{-1} \mathbf{b}, \quad (60)$$

where  $\mathbf{A}$  and  $\mathbf{b}$  are defined as

$$\mathbf{A} = \mathbf{G}_S^\top \boldsymbol{\Sigma}_S^{-1} \mathbf{W} \mathbf{P}_S \mathbf{W}^\top \boldsymbol{\Sigma}_S^{-1} \mathbf{G}_S, \quad (61)$$

$$\mathbf{b} = \mathbf{G}_S^\top \boldsymbol{\Sigma}_S^{-1} \mathbf{W} \mathbf{C}. \quad (62)$$

For maximizing Eq. (51) with respect to  $\phi$ , a gradient method can be applied using its first derivative of Eq. (59).

### 4.3 Training Procedure

The training procedure of SLT2D-HMMs can be summarized as follows:

**Step 1** Initialize the model parameters and the state sequences of SLT2D-HMMs using the parameters and

Viterbi state sequences of SL2D-HMMs, respectively.

**Step 2** Update  $\mathbf{m}$  and  $\phi$ .

**Step 3** Search sub-optimal state sequences in accordance with the procedure as summarized in Sect. 4.1.

**Step 4** If the Viterbi-approximated  $Q$ -function has not converged, return to **Step 2**. Otherwise, stop the iteration.

## 5. Experiments

### 5.1 Experimental Conditions

To demonstrate the effectiveness of the proposed model, experiments on modeling faces from the XM2VTS database [21] were conducted. The face images were extracted from the original images ( $720 \times 576$  pixels and transformed into gray-scale) and then sub-sampled to  $16 \times 16$  and  $32 \times 32$  pixels. The images of  $16 \times 16$  pixels were used for image recognition experiments and the images of  $32 \times 32$  pixels were used for state alignment experiments. Two datasets were prepared with this process:

- “dataset 1”: size-location normalized data (the original size and location in the database are used).
- “dataset 2”: data with size and location variations. The sizes and locations were randomly generated by Gaussian distributions almost within the location shift of  $40 \times 20$  pixels from the center and the range of sizes  $500 \times 500 \sim 600 \times 600$  with a fixed aspect ratio.

Figure 6 shows the examples of two datasets where the size of face image is  $16 \times 16$ . The output distribution for each state was single-Gaussian distribution. The transition probabilities for each state sequence were assumed to be a left-to-right and top-to-bottom no skip topology. The observation vectors  $\mathbf{O}$  were constructed by appending (i) the 1st order horizontal and vertical dynamic feature vectors and (ii) the 1st order horizontal, vertical and diagonal dynamic feature vectors to the static features  $\mathbf{C}$ . In the case of (ii), an observation vector  $\mathbf{O}_t$  can be constructed as

$$\mathbf{O}_t = \left[ \Delta^{(S)} \mathbf{C}_t^\top, \Delta^{(H)} \mathbf{C}_t^\top, \Delta^{(V)} \mathbf{C}_t^\top, \Delta^{(D_1)} \mathbf{C}_t^\top, \Delta^{(D_2)} \mathbf{C}_t^\top \right]^\top, \quad (63)$$

where  $\Delta^{(D_1)} \mathbf{C}_t$  and  $\Delta^{(D_2)} \mathbf{C}_t$  are diagonal dynamic feature vectors defined as

$$\Delta^{(D_1)} \mathbf{C}_t = \sum_{\tau=-L_-^{(D_1)}}^{L_+^{(D_1)}} w^{(D_1)}(\tau) \mathbf{C}_{(t^{(1)}-\tau, t^{(2)}+\tau)}, \quad (64)$$

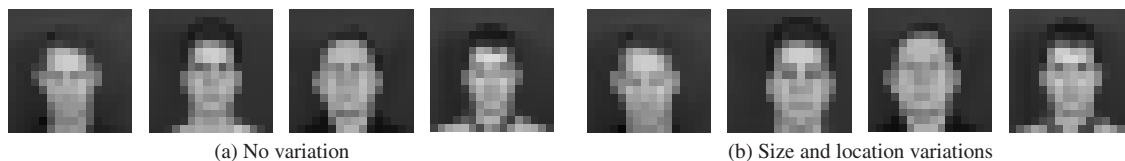
$$\Delta^{(D_2)} \mathbf{C}_t = \sum_{\tau=-L_-^{(D_2)}}^{L_+^{(D_2)}} w^{(D_2)}(\tau) \mathbf{C}_{(t^{(1)}+\tau, t^{(2)}+\tau)}. \quad (65)$$

For each case, the corresponding window matrix  $\mathbf{W}$  was designed to satisfy Eq. (23). In the case of (i),

$$L_+^{(H)} = L_-^{(H)} = L_+^{(V)} = L_-^{(V)} = 1.0, \quad (66)$$

$$w^{(H)}(-1) = w^{(V)}(-1) = -0.5, \quad (67)$$





**Fig. 6** Examples of training data; with no variation (a) and with variations of size and location (b). The size of face image is  $16 \times 16$ .

$$w^{(H)}(0) = w^{(V)}(0) = 0.0, \quad (68)$$

$$w^{(H)}(1) = w^{(V)}(1) = 0.5. \quad (69)$$

Additionally, in the case of (ii),

$$L_+^{(D_1)} = L_-^{(D_1)} = L_+^{(D_2)} = L_-^{(D_2)} = 1.0, \quad (70)$$

$$w^{(D_1)}(-1) = w^{(D_2)}(-1) = -0.5, \quad (71)$$

$$w^{(D_1)}(0) = w^{(D_2)}(0) = 0.0, \quad (72)$$

$$w^{(D_1)}(1) = w^{(D_2)}(1) = 0.5. \quad (73)$$

Although it was already confirmed that the recognition performance was significantly improved with appropriate feature vectors such as 2-D discrete cosine transform coefficients, the pixel intensity values were used as features in this paper. This is because the objective of this experiment was not to obtain the best performance of the proposed model but to demonstrate the property of the proposed model to normalize size and location variations. For the purpose of improving the recognition performance, SL2D-HMMs were extended by integrating with a linear feature extraction such as probabilistic PCA or factor analyzers [9]. In the paper, it was confirmed that SL2D-HMMs and their extensions exceed the eigenface methods and subspace methods in face recognition experiments. The structure proposed in this paper can be integrated with a linear feature extraction as [9] for improving recognition performance.

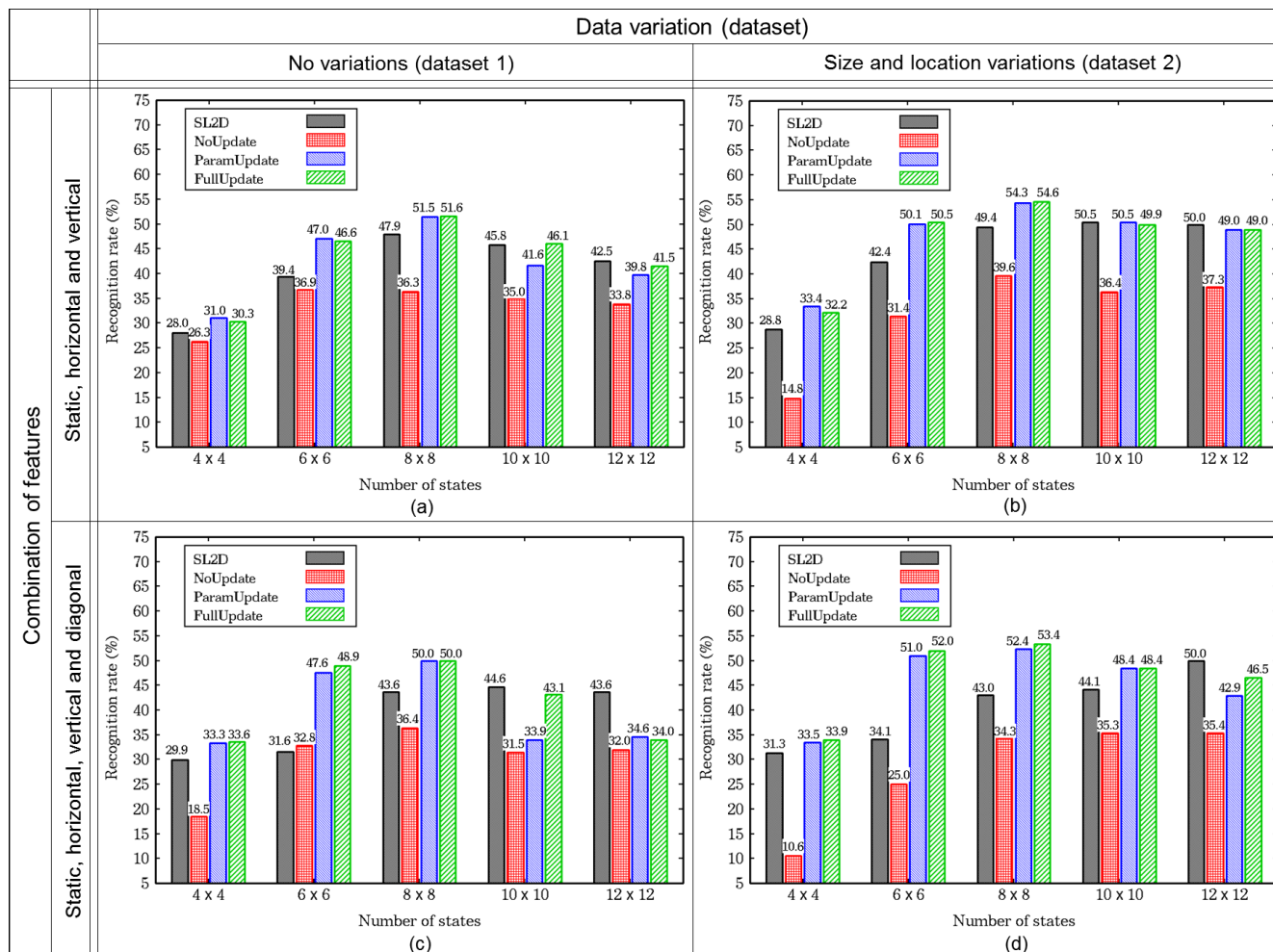
The model parameters of SLT2D-HMMs were estimated in accordance with the training procedure as summarized in Sect. 4. To make the concatenated covariance matrix  $\phi$  be positive,  $\log(\phi)$  was used in optimizing  $\phi$ , where  $\log(\cdot)$  denotes elementwise logarithm operator. The Rprop method [32], a first order gradient-based optimization method, was adopted for optimizing  $\log(\phi)$  in this paper.

## 5.2 Face Recognition Experiments

Face recognition experiments on the XM2VTS database were conducted. We prepared eight images (two images  $\times$  four sessions) of 100 subjects; six images (three sessions) were used for training and two images (remaining one session) for testing. Based on 4-fold cross validation method by alternating the sessions for training and testing, all the recognition rates were evaluated. In this experiment, the size of face images was  $16 \times 16$  and they were modeled by SL2D-HMMs and SLT2D-HMMs with  $4 \times 4$ ,  $6 \times 6$ ,  $8 \times 8$ ,  $10 \times 10$ , and  $12 \times 12$  states. Figure 7 shows recognition rates of SL2D-HMMs and SLT2D-HMMs. Figures 7 (a) and (b) show the results on “dataset1” and “dataset2,” in which 1st order horizontal and vertical dynamic features were applied, respectively. Figures 7 (c) and (d) show the results on “dataset1”

and “dataset2,” in which not only horizontal and vertical features but also diagonal features were applied, respectively. In these figures, “SL2D” means SL2D-HMMs, and “NoUpdate” means SLT2D-HMMs with the same model parameters as SL2D-HMMs, which were equivalent to the initial parameters of SLT2D-HMMs. In other words, their parameters were not optimized for SLT2D-HMMs. “ParamUpdate” means SLT2D-HMMs with the state sequences fixed, while “FullUpdate” means SLT2D-HMMs with both the model parameters and the state sequences. In “ParamUpdate” and “FullUpdate,” the initial model parameters were the same as “SL2D”.

First, the recognition rates in Fig. 7 (b) were higher than those in Fig. 7 (a) as a whole. Especially, in Fig. 7 (a), the recognition rate of 51.5% was obtained at  $8 \times 8$  states of “ParamUpdate,” while, in Fig. 7 (b), the highest recognition rate of 54.3% was obtained at the same states of “ParamUpdate.” Similar tendency could be observed from Fig. 7 (c) and Fig. 7 (d). This indicates that both SL2D-HMMs and SLT2D-HMMs could successfully reduce the influence of the variations due to the ability to normalize the size and location variations. Moreover, from our further inspection, it could be observed that the values of the variance parameters estimated from dataset 2 were bigger than that from dataset 1 as a whole. This fact suggests that the moderate variance parameters were estimated due to the size and location variations and over-fitting was slightly mitigated, and also helps to understand the reason why the recognition rates on dataset 2 were better than that on dataset 1. It can also be seen that “NoUpdate” was lower than “SL2D,” though the same model parameters were used between them. This is obviously because the parameters were not optimized for the likelihood function of SLT2D-HMMs. After the model parameters were optimized, “ParamUpdate” and “FullUpdate” achieved better results than “SL2D” and “NoUpdate.” However, when comparing “ParamUpdate” and “FullUpdate,” significant improvement of the performance could not be obtained. The reason for this result can be explained as follows: Since the observations depend on horizontal and vertical state sequences, it must be taken into account that the combinations of both state sequences affect the likelihood at the re-estimation stage for state sequences. Nevertheless, the search algorithm for state sequences as summarized in Sect. 4.1 is strongly approximated in the sense that it finds only one state boundary from all of the candidates of the horizontal and vertical state boundary at one time. Ideally, for each candidate of state boundary, small variations should be added to the other boundaries and the likelihood should be evaluated over all of these combinations. How-



**Fig. 7** Recognition rates of SL2D-HMMs and SLT2D-HMMs. Two figures on the top, (a) and (b) show the results on “dataset1” and “dataset2,” in which 1st order horizontal and vertical dynamic features were applied, respectively. On the other hand, two figures on the bottom, (c) and (d) show the results on “dataset1” and “dataset2,” in which not only horizontal and vertical features but also diagonal features were applied, respectively. The size of face image is  $16 \times 16$ . All the recognition rates were evaluated by using 4-fold cross validation method.

ever, much more computational time will be required in this strategy.

















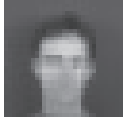


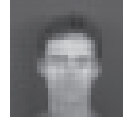

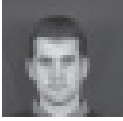

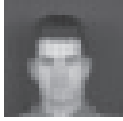




From Figs. 7 (a) and (c), it can be seen that the recognition rates in Fig. 7 (c) were slightly lower than those in Fig. 7 (a). In particular, the highest recognition rate of 50.0% at  $8 \times 8$  states of “FullUpdate” in Fig. 7 (c) was lower than that of 51.6% at the same states of “FullUpdate” in Fig. 7 (a). This is partly because the model over-fitted to the training data with size and location variations.

### 5.3 State Alignment Experiments



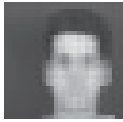


















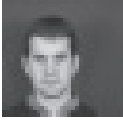
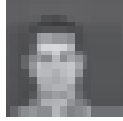
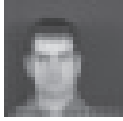
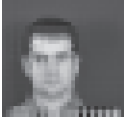



To demonstrate the advantageous property of SLT2D-HMMs for image recognition, an state alignment experiment was conducted on “dataset1” and “dataset2,” where the size of the face images was  $32 \times 32$  and the number of HMM-states was  $16 \times 16$ . Figure 8 shows the test image and its state alignments of SL2D-HMMs and SLT2D-HMMs on “dataset 1” and “dataset 2,” respectively. The alignments of

SL2D-HMMs are represented by the images that each pixel value of the input images is replaced with the mean value of the aligned states. The numerical values below the images represent the estimated log-likelihoods of the test data per pixel given the optimized state alignments. When the visualized alignment is similar to the test data, it means that the model appropriately normalized the variations of the test data. The likelihood of the test data can also be regarded as an objective measure of the similarity; higher likelihood means more preferable matching was obtained in terms of the maximum likelihood criterion.

From “SL2D” of Fig. 8 (a), it can be seen that a rectangular state alignment was obtained by using SL2D-HMMs, because of the constraint that the statistics within a state do not change dynamically. In comparison, it can be seen that the mean vector  $\bar{C}_S$  of “NoUpd” seemed smoother than the state alignment of “SL2D”. This indicates that the constraint of the SL2D-HMMs of constant statistics was mitigated. However, the detailed parts of the test data (e.g.,

test data	horizontal and vertical			horizontal, vertical and diagonal		
	SL2D	NoUpd	ParamUpd	SL2D	NoUpd	ParamUpd
	 $\mathcal{L} = -8.82$	 $\mathcal{L} = -3.03$	 $\mathcal{L} = -2.64$	 $\mathcal{L} = -14.95$	 $\mathcal{L} = -3.58$	 $\mathcal{L} = -2.79$
	 $\mathcal{L} = -8.17$	 $\mathcal{L} = -3.05$	 $\mathcal{L} = -2.75$	 $\mathcal{L} = -13.66$	 $\mathcal{L} = -3.79$	 $\mathcal{L} = -3.01$
	 $\mathcal{L} = -8.08$	 $\mathcal{L} = -2.68$	 $\mathcal{L} = -2.51$	 $\mathcal{L} = -12.86$	 $\mathcal{L} = -3.14$	 $\mathcal{L} = -2.54$
	 $\mathcal{L} = -8.03$	 $\mathcal{L} = -2.82$	 $\mathcal{L} = -2.52$	 $\mathcal{L} = -13.59$	 $\mathcal{L} = -3.57$	 $\mathcal{L} = -2.71$

(a) No variation

test data	horizontal and vertical			horizontal, vertical and diagonal		
	SL2D	NoUpd	ParamUpd	SL2D	NoUpd	ParamUpd
	 $\mathcal{L} = -9.12$	 $\mathcal{L} = -3.19$	 $\mathcal{L} = -2.85$	 $\mathcal{L} = -14.88$	 $\mathcal{L} = -3.57$	 $\mathcal{L} = -3.13$
	 $\mathcal{L} = -8.49$	 $\mathcal{L} = -3.30$	 $\mathcal{L} = -2.91$	 $\mathcal{L} = -14.04$	 $\mathcal{L} = -3.91$	 $\mathcal{L} = -3.08$
	 $\mathcal{L} = -8.52$	 $\mathcal{L} = -3.02$	 $\mathcal{L} = -2.79$	 $\mathcal{L} = -13.78$	 $\mathcal{L} = -3.51$	 $\mathcal{L} = -2.89$
	 $\mathcal{L} = -8.35$	 $\mathcal{L} = -2.81$	 $\mathcal{L} = -2.60$	 $\mathcal{L} = -13.92$	 $\mathcal{L} = -3.90$	 $\mathcal{L} = -3.03$

(b) Size and location variations

**Fig. 8** Visualization of state alignment with no variation (a) and with variations of size and location (b). “SL2D” means the state alignments of SL2D-HMMs to the test data. “NoUpd” means the mean vectors of SLT2D-HMMs without parameters optimized. “ParamUpd” means the mean vectors of SLT2D-HMMs with parameters optimized. The size of face image is  $32 \times 32$  and the number of states is  $16 \times 16$ . The  $\mathcal{L}$  means the estimated log-likelihood per pixel to test data.

eyes and nose) became blurred in “NoUpd”, since the model parameters were not optimized for SLT2D-HMMs. After the model parameters were optimized, it can be observed that the details became clearer in “ParamUpd” of Fig. 8 (a). Moreover, it can also be seen from “SL2D” of Fig. 8 (b) that SL2D-HMMs could deal with size and location variation by changing the each state duration. From “NoUpd” and “ParamUpd” of Fig. 8 (b), this property also holds true in SLT2D-HMMs. These results also explain the improvement of the recognition performance.

From both Figs. 8(a) and (b), the log-likelihoods of “ParamUpd” were higher than “NoUpd” as a whole. This fact indicates that the model parameters were optimized properly and kept the generalization ability to the test data. The one reason why the log-likelihoods of “SL2D” were lower than that of “NoUpd” and “ParamUpd” on the whole was that the constant statistics within each state of SL2D-HMMs. The another reason was that the observation vectors  $O$  in SL2D-HMMs were composed of the static and dynamic features, while the observation vectors in SLT2D-HMMs were the only static features  $C$ . Since the negative log-likelihood to the test data represents roughly the squared error between the test data and aligned mean vectors considering the covariance, the error itself will be increased by augmenting the dimensionality of the observation. As a result, this leads to an decrease in the likelihood of SL2D-HMMs. The fact that the log-likelihoods of “SL2D” in Figs. 8 (a) and (b) on the right side (horizontal, vertical and diagonal) were lower than that on the left side (horizontal and vertical) also follows the same reason.

## 6. Conclusion

In this paper, a novel statistical model based on 2-D HMMs for image recognition was proposed. It has been known that SL2D-HMMs have shortcomings inherited from standard HMMs, that is, the stationary statistics within each state and the conditional independent assumption of state output probabilities. To overcome these shortcomings of SL2D-HMMs, the proposed model can be derived by reformulating SL2D-HMMs and imposing explicit relationships between static and dynamic features. As a result, the proposed model can capture the dependencies of adjacent observations, without increasing the number of model parameters. Experiments on image recognition and state alignment were conducted on the XM2VTS database. The proposed model achieved better results than SL2D-HMMs.

For future work, we are planning to append not only 1st order dynamic features, but also more higher order dynamic features. Implementing more precise search algorithms such as the delayed decision Viterbi algorithm [20] will also be future work.

## References

- [1] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of IEEE*, vol.77, pp.257–285, 1989.
- [2] F. Samaria and F. Fallside, “Face identification and feature extraction using hidden markov models,” *Image Processing: Theory and Applications*, pp.295–298, Elsevier, 1993.
- [3] S. Kuo and O. Agazzi, “Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.16, pp.842–848, 1994.
- [4] A.V. Nefian and M.H. Hayes III, “Maximum likelihood training of embedded HMM for face detection and recognition,” *IEEE International Conference on Image Processing (ICIP)*, vol.25, no.10, pp.1229–1238, 2003.
- [5] H. Othman and T. Aboulnasr, “A simplified second-order HMM with application to face recognition,” *International Symposium on Circuits and Systems*, vol.2, pp.161–164, 2001.
- [6] J.T. Chien and C.P. Liao, “Maximum confidence hidden Markov modeling for face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.30, pp.606–616, 2008.
- [7] J. Li, A. Najmi, and R.M. Gray, “Image classification by a two-dimensional hidden Markov model,” *IEEE Trans. Signal Process.*, vol.48, no.2, pp.517–533, 2000.
- [8] D. Kurata, Y. Nankaku, K. Tokuda, T. Kitamura, and Z. Ghahramani, “Face recognition based separable lattice HMMs,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp.737–740, 2006.
- [9] Y. Nankaku and K. Tokuda, “Face recognition based hidden Markov eigenface models,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp.469–472, 2007.
- [10] A. Tamamori, Y. Nankaku, and K. Tokuda, “An extension of separable lattice 2-D HMMs for rotational data variations,” *IEICE Trans. Inf. & Syst.*, vol.E95-D, no.8, pp.2074–2083, Aug. 2012.
- [11] Y. Takahashi, A. Tamamori, Y. Nankaku, and K. Tokuda, “Face recognition based on separable lattice 2-D HMMs with state duration control,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp.2162–2165, 2010.
- [12] K. Kumaki, Y. Nankaku, and K. Tokuda, “Face recognition based on extended separable lattice 2-D HMMs,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol.25, pp.2209–2212, 2012.
- [13] S. Uchida and H. Sakoe, “An approximation algorithm for two-dimensional warping,” *IEICE Trans. Inf. & Syst.*, vol.E83-D, no.1, pp.109–111, Jan. 2000.
- [14] S. Uchida and H. Sakoe, “Piecewise linear two-dimensional warping,” *Systems and Computers in Japan*, vol.32, no.12, pp.1–9, 2001.
- [15] N. Suto, T. Nishimura, R.H. Fujii, and R. Oka, “Spotting recognition of concave and convex reference image with pixel-wise correspondence using two-dimensional continuous dynamic programming,” *IEICE Technical Reports*, vol.103, no.210, pp.23–28, 2003.
- [16] Y. Yaguchi, K. Iseki, N.T. Viet, and R. Oka, “Full pixel matching between images for non-linear registration of objects,” *IPSI Transactions on Computer Vision and Applications*, vol.2, pp.1–14, 2010.
- [17] S. Nakagawa, “A survey on automatic speech recognition,” *IEICE Trans. Inf. & Syst.*, vol.E85-D, no.3, pp.465–486, March 2002.
- [18] S. Furui, “Speaker independent isolated word recognition using dynamic features of speech spectrum,” *IEEE Trans. Acoust. Speech Signal Process.*, vol.ASSP-34, pp.52–59, 1986.
- [19] K. Kumaki, *Face Recognition based on Extended Separable Lattice HMMs*, Bachelor thesis, Nagoya Institute of Technology, 2010.
- [20] H. Zen, K. Tokuda, and T. Kitamura, *Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences*, Ph.D. thesis, Nagoya Institute of Technology, 2006.
- [21] K. Messer, J. Mates, J. Kitter, J. Luetten, and G. Maitre, “XM2VTS:

The extended M2VTS database,” Proceedings of Audio and Video-Based Biometric Person Authentication, pp.72–77, 1999.

- [22] Z. Ghahramani, M.I. Jordan, and P. Smyth, “Factorial hidden markov models,” in *Machine Learning*, pp.245–293, MIT Press, 1997.
- [23] C.H. Lee and E. Giachin, “Speaker independent isolated word recognition using dynamic features of speech spectrum,” IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol.1, pp.161–164, 1991.
- [24] H. Zen, Y. Nankaku, K. Tokuda, and T. Kitamura, “Product of experts for statistical parametric speech synthesis,” IEEE Trans. Audio, Speech and Language Processing, vol.20, no.3, pp.153–173, March 2012.
- [25] G. Hinton, “Product of experts,” International Conference on Artificial Neural Networks (ICANN), vol.1, pp.1–6, 1999.
- [26] M. Gales and S. Airey, “Product of Gaussians for speech recognition,” tech. rep., CUED/F-INFENG/TR.458, 2003.
- [27] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, “Adaptive mixtures of local experts,” *Neural Computation*, vol.3, no.1, pp.79–87, 1991.
- [28] D.A. Reynolds and R.C. Rose, “Robust text-independent speaker identification using Gaussian mixture speaker models,” IEEE Trans. Speech Audio Process., vol.3, no.1, pp.72–83, 1995.
- [29] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, Chapman & Hall/CRC, 2005.
- [30] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of Royal Statistics Society*, vol.39, pp.1–38, 1977.
- [31] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimal decoding algorithm,” IEEE Trans. Inf. Theory, vol.IT-13, pp.260–269, 1967.
- [32] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: The Rprop algorithm,” Proceedings of IEEE ICNN, pp.586–591, 1993.



**Akira Tamamori** received the B.E. degree in Computer Science, and his M.E. degrees in the Department of Scientific and Engineering Simulation from Nagoya Institute of Technology, Nagoya, Japan, in 2008, 2010 respectively. He is currently a Ph.D. candidate at the same institute. His research interests include statistical machine learning, image recognition, and speech signal processing.



**Yoshihiko Nankaku** received the B.E. degree in Computer Science, and his M.E. and Ph.D. degrees in the Department of Electrical and Electronic Engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1999, 2001, and 2004 respectively. After a year as a postdoctoral fellow at the Nagoya Institute of Technology, he became an Associate Professor at the same Institute. He was a visiting researcher at the Department of Engineering, University of Cambridge, UK, from May to October

2011. His research interests include statistical machine learning, speech recognition, speech synthesis, image recognition, and multi-modal interface. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and the Acoustical Society of Japan (ASJ).



**Keiichi Tokuda** received the B.E. degree in electrical and electronic engineering from Nagoya Institute of Technology, Nagoya, Japan, his M.E. and Dr.Eng. degrees in information processing from Tokyo Institute of Technology, Tokyo, Japan, in 1984, 1986, and 1989, respectively. From 1989 to 1996 he was a Research Associate at the Department of Electronic and Electric Engineering, Tokyo Institute of Technology. From 1996 to 2004 he was an Associate Professor at the Department of Computer Science,

Nagoya Institute of Technology as Associate Professor, and now he is a Professor at the same institute. He is also an Invited Researcher at ATR Spoken Language Translation Research Laboratories, Japan and was a Visiting Researcher at Carnegie Mellon University, Pittsburgh, PA, from 2001 to 2002. He published over 70 journal papers and over 200 conference papers, and received 5 paper awards. He was a member of the Speech Technical Committee of the IEEE Signal Processing Society from 2000 to 2003. He is currently a member of ISCA Advisory Council and an associate editor of IEEE Transactions on Audio, Speech & Language Processing, and acts as organizer and reviewer for many major speech conferences, workshops and journals. His research interests include speech coding, speech synthesis and recognition, and statistical machine learning.