DOCTORAL DISSERTATION

STATISTICAL MODELS INCLUDING NORMALIZATION PROCESSES FOR IMAGE RECOGNITION
(画像認識のための正規化プロセスを含んだ統計モデル)

DOCTOR OF ENGINEERING

DECEMBER 2014

Akira TAMAMORI

Supervisor : Dr. Keiichi TOKUDA
Dr. Akinobu LEE

Department of Scientific and Engineering Simulation
Nagoya Institute of Technology

Abstract

For many years, many researchers of pattern recognition have developed the field of image recognition as the main focus of pattern recognition and various techniques have been proposed. Especially, statistical approaches based on Principal Component Analysis (PCA) such as eigenface methods and subspace methods show good recognition performance in many applications. However, if images contain geometric variations such as size, location and rotation, the recognition performance is significantly degraded. Therefore, normalization processes for such geometric variations are required prior to applying these methods.

In many image recognition systems, the normalization process is included in the preprocess part of the classification, and heuristic normalization techniques are used. However, it is necessary to develop the normalization technique for each task, because such heuristic techniques usually use task dependent information. Furthermore, in image recognition, the final objective is not to accurately normalize images for human perception but to achieve a better recognition performance. Therefore, it is natural to use the same criterion for both training classifiers and normalization. This means that the normalization process should be integrated into classifiers.

HMM based techniques for image recognition have been proposed to reduce the influence of geometric variations. Geometric matching between input images and model parameters is represented by discrete hidden variables, and the normalization process is included in calculating probabilities. However, the extension of HMMs to multi-dimensions generally leads to an exponential increase in the amount of computation for its training algorithm. To deal with this problem, separable lattice 2-D HMMs (SL2D-HMMs) have been proposed to reduce computational complexity while retaining good properties that model multi-dimensional data. 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. However, the modeling accuracy is still insufficient because of the following problems:

i) SL2D-HMMs cannot represent rotational variations. Therefore, affine deformation cannot be modeled completely.

ii) The statistics of each state do not change dynamically.

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

In this dissertation, statistical models to improve the recognition performance which can overcome the above problems of SL2D-HMMs are proposed.

First, a new generative model which can deal with rotational data variations is proposed, by extending SL2D-HMMs. To reduce the complexity, SL2D-HMMs have only one state sequence in each direction; this means that all horizontal/vertical lines of an observation lattice have the same state alignment for each direction. However, to represent the rotational variations, the models should have a different state alignment for each observation line and horizontal/vertical state alignments should be changed along with vertical/horizontal direction. Furthermore, it should take account of the dependency of the state alignments between consecutive observation lines to perform a continuous elastic matching. In this paper, we introduce additional HMM states which represent the shifts of the state alignments of the observation lines in a particular direction. In face recognition experiments the proposed model achieved better results to the images than the conventional SL2D-HMMs. Moreover, the state alignments shows that the proposed model can normalize not only size and location variations but also rotational variations.

Furthermore, a novel statistical model based on 2-D HMMs is proposed to overcome the shortcomings of ii) and iii). Although these are the essential assumption inherited from 1-D HMMs, 1-D trajectory HMMs were proposed and successfully applied to speech recognition and speech synthesis, which can overcome the shortcomings of 1-D HMMs. This dissertation derives 2-D trajectory HMMs by reformulating the likelihood of SL2D-HMMs with imposing explicit relationships between static and dynamic features. The proposed model can overcome the shortcomings of ii) and iii) and efficiently capture dependencies between adjacent observations without increasing the number of model parameters. Experimental results show that the proposed model achieved better recognition performance than the conventional SL2D-HMMs.

Keywords: image recognition, hidden Markov model, separable lattice 2-D HMMs, trajectory HMMs.

Abstract in Japanese

長年に渡り，画像認識の分野はパターン認識における主要なテーマとして多くの研究者の手によって発展し，様々な手法が提案されてきた．とくに固有顔法や部分空間法に代表される主成分分析に基づく統計的なアプローチは良好な認識性能を持つことが多くの応用分野で示されている．しかしながら，認識対象の画像が位置や大きさ，回転のような幾何学的変動を含む場合，認識性能が大きく低下する．それゆえ，これらの手法を適用する前にはそのような幾何学的変動に関する正規化プロセスが必要である．多くの画像認識システムでは正規化プロセスは識別の前処理部に含まれ，そこではヒューリスティックな正規化手法が用いられる．しかし，そのような正規化手法はタスク依存の情報を用いることが通常であるので，各タスクに関して正規化手法を考える必要がある．そのうえ，画像認識においては，最終的な目標というのは人間の知覚に関し画像を正規化するだけでなく，良好な認識性能を達成することである．ゆえに識別器と正規化の両方で共通の基準を用いるのが自然である．このことは正規化プロセスが識別器に統合されるべきである，ということを意味する．

そのような幾何学的な変動に対するアプローチとして，隠れマルコフモデル(Hidden Markov Model; HMM)に基づく手法が近年提案されている．この手法では入力画像とモデルパラメータとの幾何学的なマッチングは離散的な隠れ変数により表現され，正規化プロセスはそれらにおける確率の計算に含まれる．しかしHMMを多次元に拡張する場合，一般的に学習アルゴリズムに関して計算量が指数的に増大するという問題がある．そこで計算量を削減しつつ多次元データの良い性質を保つための手法として，近年分離型格子2次元HMM(Separable Lattice 2-D HMM; SL2D-HMM)が提案されている．SL2D-HMMは縦・横2つの隠れ状態系列からなる構造を有し，それら系列は格子点上での観測のモデル化において相互に影響し合う．SL2D-HMMは水平方向と垂直方向で柔軟なマッチングを行うことができるため，対象の位置や大きさに関する不変量だけでなく各方向での非線形なひずみをモデル化することができるという利点がある．しかしながら，SL2D-HMMには以下に述べるような問題点が存在するため，モデル化の精度は依然として不十分である．

1. モデル構造の制約のために，回転変動を表現することができない．したがって，

アフィン変換を完全にモデル化することができない.

2. 各状態内で統計量が定常的である．すなわち，各状態内では出力確率分布が一定であり，同じ状態内で動的に変化する観測を捉えることが困難ということである.

3. 各時刻における観測ベクトルの出力確率は，その時刻に滞在する状態にのみ依存し，前後の時刻に滞在する状態には依存しない．これは，独立性の仮定 (Conditional Independence Assumption) と呼ばれる.

本論文では，SL2D-HMM の上記の問題点を克服した画像認識のためのより高性能な統計モデルの提案を目的とする.

まず，本論文では SL2D-HMM を拡張し，回転変動にも対応可能となる新たな生成モデルの構造を提案する．SL2D-HMM では計算量を削減するため各方向で状態系列は 1 つだけであった．これは格子点上の観測において全ての水平 (垂直) ライン上で同一の状態アラインメントを有するということを意味する．しかし回転変動を表現するためにはモデルはそれらライン上で異なる状態アラインメントを有するべきであり，水平 (垂直) 方向の状態アラインメントは垂直 (水平) 方向に沿って変化すべきである．さらに連続的で柔軟なマッチングを行うためには，一連の観測ライン間で状態アラインメントの依存関係を考慮に入れる必要がある．そこで本研究では，ある特定方向での観測ラインに関する状態アラインメントのシフトを表現する HMM 状態系列を新たに導入する．これにより，位置・大きさの変動だけでなく回転変動にも対応可能となることが期待される．位置や大きさの変動だけでなく，回転変動を含む画像認識実験の結果，提案モデルは SL2D-HMM と比較して良好な認識性能を持つことが示された．さらに，状態アライメントを可視化することで，提案モデルは位置や大きさだけでなく，回転変動を正しく正規化できることを確認した.

さらに，上記問題点 2. と 3. を同時に克服する新たな統計モデルを提案する．これらは 1 次元 HMM から継承される本質的な仮定であるが，すでに 1 次元 HMM の場合では問題解決のためにトラジェクトリ HMM が提案されており，音声認識や音声合成に適用され成功を収めている．本論文では，静的及び動的特徴を含む特徴ベクトルを状態出力ベクトルとする SL2D-HMM に対して，静的・動的特徴間の関係を明示的に導入し，SL2D-HMM を再定式化する．結果として，2 次元のトラジェクトリ HMM を導出可能であることを示す．提案モデルは上記 SL2D-HMM の問題点を避けることができる．また，提案モデルのモデルパラメータの数は SL2D-HMM のパラメータ数から増加することはないので，隣接する観測間の相関を効率よく捉えることが可能となる．画像認識実験の結果より，提案モデルは SL2D-HMM よりも良好な認識性能を持つことが示された.

以上のように，本論文では，統計的手法による画像認識のための汎用的かつ，より

高精度なモデル化手法を提案し，これらの手法の有効性を示す.

Acknowledgment

First of all, I would like to express my sincere gratitude to Keiichi Tokuda, my advisor, for his support, encouragement, and guidance.

I would like to thank Yoshihiko Nankaku, Akinobu Lee, Keiichiro Oura, and Kei Hashimoto for their technical supports and helpful discussions. Special thanks go to all the members of Tokuda and Lee laboratories for their technical support and encouragement. If somebody was missed among them, my work would not be completed. I would be remiss if I did not thank Natsuki Kuromiya and Masayo Fujimura, secretaries of the laboratory, for their kind assistance.

Finally, I would sincerely like to thank my parents and my friends for their encouragement.

Contents

List of Figures **ix**

1 Introduction **2**

2 Hidden Markov Models **6**

2.1 Definition of HMM 6

2.2 Calculation of output probability 8

2.2.1 Total output probability of an observation vector sequence 8

2.2.2 Forward-Backward algorithm 9

2.3 Searching optimal state sequence 11

2.4 Maximum likelihood estimation of HMM parameters 12

2.4.1 Q-function 12

2.4.2 Maximization of Q-function 13

2.5 Summary 16

3 Separable Lattice 2-D Hidden Markov Models **17**

3.1 Related work 17

3.2 Model definition 20

3.3 Training algorithm 23

3.3.1 EM algorithm 23

3.3.2 Variational EM algorithm 25

3.3.3 Variational DAEM algorithm 28

3.4 Disadvantage 30

3.5 Summary 30

4 An extension of separable lattice 2-D HMMs for rotational variations 32

4.1 Model structure representing rotational variations 32

4.2 Training algorithm 36

4.3 Experiments 40

4.3.1 Experimental conditions 40

4.3.2 Experimental results 42

4.4 Summary 44

5 Separable lattice trajectory 2-D HMMs 48

5.1 Applications of dynamic features 49

5.1.1 Dynamic features for speech data 49

5.1.2 Dynamic features for image data 50

5.2 Model definition 52

5.3 Relation to other statistical models 55

5.4 Training algorithm 56

5.4.1 Estimation of sub-optimum state sequence 56

5.4.2 Estimation of model parameters 57

5.4.3 Training procedure 58

5.5 Experiments 59

5.5.1 Experimental conditions 59

5.5.2 Face recognition experiments 60

5.5.3 State alignment experiments 63

5.6 Summary 64

6 Conclusions 66

6.1 Summary 66

6.2 Future work 67

List of Publications 73

Journal dissertations 73

International conference proceedings 73

Technical reports 74

Domestic conference proceedings 74

Appendix A Derivation of training algorithm for an extension of separable lattice 2-D HMMs for rotational variations 75

A.1 Derivation of approximated posterior distributions 75

A.1.1 Detail of approximated posterior distributions 77

A.2 Derivation of re-estimation formulas 80

Appendix B Derivation separable lattice trajectory 2-D HMMs 82

Appendix C Derivation of reestimation formula for the concatenated mean vector *m* 84

List of Figures

2.1 Examples of HMM structure. 7

2.2 Implementation of the computation using forward-backward algorithm in terms of a trellis of observations and states. 10

3.1 Graphical model representation of the pseudo 2-D HMMs: 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. 19

3.2 Model structure of the separable lattice 2-D HMMs: hidden state sequences are composed of independent two Markov chains. 21

3.3 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. 22

4.1 Model structure of the proposed model: The horizontal/vertical state alignments is changed along with vertical/horizontal state direction to represent the rotational variations. 34

4.2 Graphical representation of the proposed model: The shift sequence affects the all data on the same observed line. 35

4.3 An example of state alignment of the proposed model for reference states and shifted states in horizontal direction: Without shift states (SL2D-HMMs), rectangle state alignments can be obtained while with shift states, monotonically shifted state alignments can be obtained in the proposed model. 36

4.4 The example of topology of the transition probabilities of the m-th dimension shift states where $D^{(m)}_{min} = -2$ and $D^{(m)}_{max} = 2$; from this topology, monotonically increasing or decreasing sequence of the shift amount can be obtained and clockwise or counterclockwise rotational variations can be represented. 37

4.5 Examples of training data; with no variation (a) and with variations of size and location (b), with rotational variations (c) and with variations of size, location and rotations (d). 41

4.6 Recognition rates of the SL2D-HMMs and proposed model for each shift states tested on the dataset with no variation (a), with variations of size and location (b), with rotational variations (c) and with variations of size, location and rotations (d), respectively. In the figures, plain boxes and meshed ones represent the recognition rates of the models trained from the dataset with no variation and the same variation as the test dataset, respectively. 43

4.7 Example of mean vectors: (i) is the mean vectors of the SL2D-HMMs. (ii) is the mean vectors of the proposed model. The number of shift state of (ii) is 22×22. They were estimated from the normalized data ("dataset 1"). 45

4.8 Examples of test data and the visualized state alignments on the dataset with no variation (a), with variations of size and location (b), with rotational variations (c) and with variations of size, location and rotations (d), respectively.The θ means the rotational angle for each test data. The $\mathcal{F}$ means the estimated log-likelihood to test data. 46

5.1 An example of relationship between the observation vector sequence $\boldsymbol{o}$ and the static feature vector sequence $\boldsymbol{c}$ in a matrix form [1], where the dynamic feature vectors are calculated using Eqs. (5.3) and (5.4) with $L^{(1)}_{-} = L^{(1)}_{+} = L^{(2)}_{-} = L^{(2)}_{+} = 1$, $w^{(1)}(-1) = -0.5$, $w^{(1)}(0) = 0.0$, $w^{(1)}(1) = 0.5$, $w^{(2)}(-1) = 1.0$, $w^{(1)}(0) = -2.0$, $w^{(2)}(1) = 1.0$. 50

5.2 Examples of $\boldsymbol{w}_t^{(S)}$, $\boldsymbol{w}_t^{(H)}$, and $\boldsymbol{w}_t^{(V)}$, 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. (5.11) and (5.12). 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. (5.18), (5.19), and (5.20). 52

5.3 Examples of covariance matrix. (a) shows the covariance matrix $\boldsymbol{\Sigma_S}$ of SL2D-HMMs in Eq. (5.24) and (b) shows the covariance matrix $\boldsymbol{P_S}$ of SLT2D-HMMs in Eq. (5.28) 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 32×32. The rows and columns are aligned in raster order of the 2-D lattice (see Fig. 3.3). 53

5.4 Examples of training data; with no variation (a) and with variations of size and location (b). The size of face image is 16×16. 61

5.5 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 16×16. 62

5.6 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 32×32 and the number of states is 16×16. The $\mathcal{L}$ means the estimated log-likelihood per pixel to test data. 65

Chapter 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 [2]. 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-dimentional 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 [3–13]. 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 [3]. 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 [4] (embedded HMMs [5]) 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 reduces the computational complexity and the maximum likelihood training algorithm has been proposed [6]. 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 [14–17], 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 [9] to reduce computational complexity while retaining good properties that model multi-dimensional data. Furthermore, hidden Markov eigenface models have been proposed [10] 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 non-linear warping in all dimensions. However, the modeling accuracy is still insufficient because of the following problems:

i) SL2D-HMMs cannot represent rotational variations. Therefore, affine deformation cannot be modeled completely.

ii) The statistics of each state do not change dynamically.

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

In the present dissertation, statistical models to improve the recognition performance which can overcome the above problems of SL2D-HMMs are proposed.

First, a new generative model which can deal with rotational data variations by extending SL2D-HMMs. To reduce the complexity, SL2D-HMMs have only one state sequence in each direction; this means that all horizontal/vertical lines of an observation lattice have the same state alignment for each direction. However, to represent the rotational variations, the models should have a different state alignment for each observation line and horizontal/vertical state alignments should be changed along with vertical/horizontal direction. Furthermore, it should take account of the dependency of the state alignments between consecutive observation lines to perform a continuous elastic matching. In this paper, we introduce additional HMM states which represent the shifts of the state alignments of the observation lines in a particular direction. The parameters of this proposed model can be estimated via the expectation maximization (EM) algorithm for approximating the Maximum Likelihood (ML) estimate. However, similar to the training of SL2D-HMMs, the exact expectation step is computationally intractable. To derive a feasible algorithm, we applied the variational EM algorithm [18] to the our proposed model. The variational method approximates the posterior distribution over the hidden variables by a tractable distribution.

Furthermore, in the present dissertation, we derive a novel statistical model based on SL2D-HMMs to overcome their shortcomings. 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 [19] 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) [20] 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 [13, 21], 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 [1], 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 present dissertation is organized as follows. The next chapter 3 introduces basic theories of the 1-D HMM and also describes the model structure of SL2D-HMMs and their training algorithms based on the EM algorithm and variational EM algorithm. Chapter 5 extends the model structure of SL2D-HMMs for rotational variations and derives the training algorithm based on variational EM algorithm. Chapter 5 reformulates SL2D-HMMs by imposing explicit relationship between static and dynamic features and defines SLT2D-HMMs. Relationships between SLT2D-HMMs and other techniques are also discussed in this chapter. The training algorithm for SLT2D-HMMs is also described in this chapter. Concluding remarks and future plans are presented in the final chapter.

Chapter 2

Hidden Markov Models

Hidden Markov models (HMMs) are one of widely used statistical models for representing time series by well-defined algorithms. They have successfully been applied to acoustic modeling both in speech recognition and synthesis. This section describes its basic theories, how to calculate output probabilities of an observation vector sequence, and how to estimate its parameters.

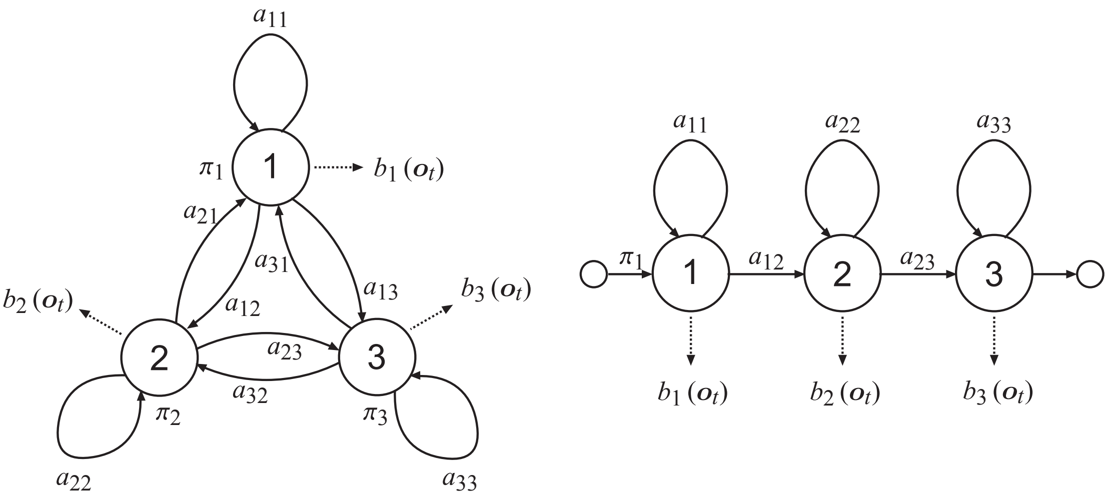
2.1 Definition of HMM

An HMM [22–24] is a finite state machine which generates a sequence of discrete time observations. At each frame it changes states according to its state transition probability distributions, and then generates an observation at time t, $\boldsymbol{O}_t$, according to its output probability distribution of the current state. Therefore, the HMM is a doubly stochastic random process model.

An N-state HMM consist of state transition probability distributions $\left\{a_{ij}\right\}_{i,j=1}^{N}$, output probability distributions $\left\{b_j\left(\boldsymbol{O}_t\right)\right\}_{j=1}^{N}$, and initial state probability distributions $\{\pi_i\}_{i=1}^{N}$. For convenience, the compact notation is used to indicate the parameter set of the model Λ as follows:

$$\Lambda = \left[\left\{a_{ij}\right\}_{i,j=1}^{N},\ \left\{b_j\left(\cdot\right)\right\}_{j=1}^{N},\ \{\pi_i\}_{i=1}^{N}\right] \tag{2.1}$$

Figure 2.1 shows examples of the HMM structure. Figure 2.1(a) shows a 3-state ergodic model, in which every state of the model could be reached from every state of the model in a single step, and Figure 2.1(b) shows a 3-state left-to-right model, in which the state



(a) A 3-state ergodic model

(b) A 3-state left-to-right model

Figure 2.1: Examples of HMM structure.

index increases or stays the same state as time increases. The left-to-right HMMs are generally used to model speech parameter sequences, since they can appropriately model signals.

The output probability distributions $\left\{b_j(\cdot)\right\}_{j=1}^{N}$ can be discrete or continuous depending on the observations. In continuous distribution HMM (CD-HMM), each output probability distribution is usually modeled by a mixture of multivariate Gaussian components [25] as follows:

$$b_j(\boldsymbol{O}_t) = \sum_{m=1}^{M} w_{jm} \cdot \mathcal{N}\left(\boldsymbol{O}_t \mid \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right), \tag{2.2}$$

where M, w_{jm}, $\boldsymbol{\mu}_{jm}$, and $\boldsymbol{\sigma}_{jm}$ are the number of Gaussian components, the mixture weight, mean vector, and covariance matrix of the m-th Gaussian component of the j-th state, respectively. Each Gaussian component is defined by

$$\mathcal{N}\left(\boldsymbol{O}_t \mid \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}\right) = \frac{1}{\sqrt{(2\pi)^K \left|\boldsymbol{\Sigma}_{jm}\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{O}_t - \boldsymbol{\mu}_{jm}\right)^{\top} \boldsymbol{\Sigma}_{jm}^{-1}\left(\boldsymbol{O}_t - \boldsymbol{\mu}_{jm}\right)\right\}, \tag{2.3}$$

where symbol $\top$ means transpose of vector or matrix, and K is the dimensionality of an observation vector $\boldsymbol{O}_t$. For each state, $\left\{w_{jm}\right\}_{m=1}^{M}$ should satisfy the stochastic constraint

$$\sum_{m=1}^{M} w_{jm} = 1, \quad 1 \leq j \leq N \tag{2.4}$$

$$w_{jm} \geq 0, \quad \begin{array}{l} 1 \leq j \leq N \\ 1 \leq m \leq M \end{array} \tag{2.5}$$

so that $\left\{b_j(\cdot)\right\}_{j=1}^{N}$ are properly normalized, i.e.,

$$\int_{\mathrm{R}^K} b_j(\boldsymbol{O}_t)\, d\boldsymbol{O}_t = 1. \quad 1 \leq j \leq N \tag{2.6}$$

It is noted that due to the model structure of HMMs, HMMs have the constraints [19] 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.

2.2 Calculation of output probability

2.2.1 Total output probability of an observation vector sequence

When a state sequence is determined, a joint probability of an observation vector sequence $\boldsymbol{O} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_T\}$ and a state sequence $\boldsymbol{S} = \{s_1, s_2, \ldots, s_T\}$ is calculated by multiplying the state transition probabilities and state output probabilities for each state, that is,

$$P(\boldsymbol{O}, \boldsymbol{S} \mid \Lambda) = \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(\boldsymbol{O}_t), \tag{2.7}$$

where $a_{s_0 s_1}$ denotes π_{s_1}. The total output probability of the observation vector sequence from the HMM is calculated by marginalizing Eq. (2.7) over all possible state sequences,

$$P(\boldsymbol{O} \mid \Lambda) = \sum_{\text{all } \boldsymbol{q}} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(\boldsymbol{O}_t). \tag{2.8}$$

The order of $2T \cdot N^T$ calculation is required, since at every $t = 1, 2, \ldots, T$ there are N possible states that can be reached (i.e., there are N^T possible state sequences). This calculation is computationally infeasible, even for small values of N and T; e.g., for

$N = 5$ (states), $T = 100$ (observations), there are on the order of $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ computations. Fortunately, there is an efficient algorithm to calculate Eq. (2.8) using forward and backward procedures.

2.2.2 Forward-Backward algorithm

The forward-backward algorithm is generally used to calcurate $P(\boldsymbol{O} \mid \Lambda)$, which is the probability of the observation sequence $\boldsymbol{O}$ given the model Λ. If I directly calculate $P(\boldsymbol{O} \mid \Lambda)$, it requires on the order of $2T \cdot N^T$ calculation. The detail of the forward-backward algorithm is described in the following part.

The probability of a partial observation vector sequence from time 1 to t and the i-th state at time t, given the HMM Λ is defined as

$$\alpha_t(i) = P(\boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_t, s_t = i \mid \Lambda). \tag{2.9}$$

$\alpha_t(i)$ is calculated recursively as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{O}_1), \quad 1 \leq i \leq N \tag{2.10}$$

2. Recursion

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(\boldsymbol{O}_t), \qquad \begin{array}{l} 1 \leq j \leq N \\ t = 2, \ldots, T \end{array} \tag{2.11}$$

3. Termination

$$P(\boldsymbol{O} \mid \Lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{2.12}$$

As the same way as the forward algorithm, backward variables $\beta_t(i)$ are defined as

$$\beta_t(i) = P(\boldsymbol{O}_{t+1}, \boldsymbol{O}_{t+2}, \ldots, \boldsymbol{O}_T \mid s_t = i, \Lambda), \tag{2.13}$$

that is, the probability of a partial vector observation sequence from time t to T, given the i-th state at time t and the HMM Λ. The backward variables can also be calculated in a recursive manner as follows:

1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{2.14}$$

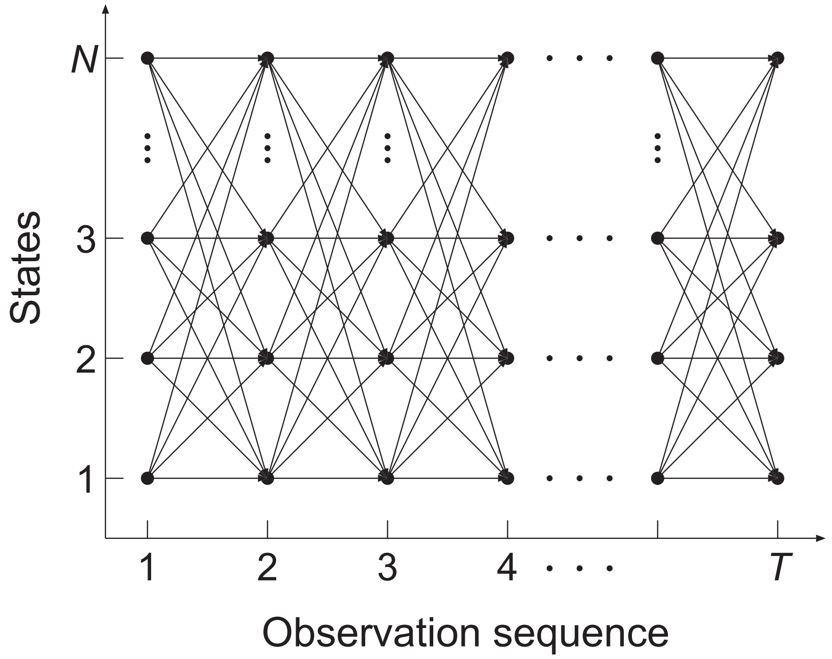


Figure 2.2: Implementation of the computation using forward-backward algorithm in terms of a trellis of observations and states.

2. Recursion

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{O}_{t+1}) \beta_{t+1}(j), \qquad \begin{array}{l} 1 \leq i \leq N \\ t = T-1, \ldots, 1. \end{array} \tag{2.15}$$

3. Termination

$$P(\boldsymbol{O} \mid \Lambda) = \sum_{i=1}^{N} \beta_1(i). \tag{2.16}$$

The forward and backward variables can be used to compute the total output probability as follows:

$$P(\boldsymbol{O} \mid \Lambda) = \sum_{j=1}^{N} \alpha_t(j) \beta_t(j). \quad 1 \leq t \leq T \tag{2.17}$$

The forward-backward algorithm is based on the trellis structure shown in Figure 2.2. In this figure, the x-axis and y-axis represent observations and states of an HMM, respectively. On the trellis, all possible state sequences will re-merge into these N nodes no matter how long the observation sequence. In the case of the forward algorithm, at time $t = 1$, I need to calculate values of $\alpha_1(i)$, $1 \leq i \leq N$. At times $t = 2, 3, \ldots, T$, I need only calculate values of $\alpha_t(j)$, $1 \leq j \leq N$, where each calculation involves only the N previous values of $\alpha_{t-1}(i)$ because each of the N grid points can be reached from only the N grid points at the previous time slot. As a result, the forward-backward algorithm can reduce order of probability calculation.

2.3 Searching optimal state sequence

The single optimal state sequence $\hat{\boldsymbol{S}} = \{\hat{s_1}, \hat{s_2}, \ldots, \hat{s_T}\}$ for a given observation vector sequence $\boldsymbol{O} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_T\}$ is useful for various applications (e.g., decoding, initializing HMM parameters). By using a manner similar to the forward algorithm, which is often referred to as the Viterbi algorithm [26], I can obtain the optimal state sequence $\hat{\boldsymbol{S}}$. Let $\delta_t(i)$ be the likelihood of the most likely state sequence ending in the i-th state at time t

$$\delta_t(i) = \max_{s_1,\ldots,s_{t-1}} P\left(s_1, \ldots, s_{t-1}, s_t = i, \boldsymbol{O}_1, \ldots, \boldsymbol{O}_t \mid \Lambda\right), \tag{2.18}$$

and $\psi_t(i)$ be the array to keep track. The complete procedure for finding the optimal state sequence can be written as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\boldsymbol{O}_1), \qquad 1 \leq i \leq N \tag{2.19}$$

$$\psi_1(i) = 0, \qquad 1 \leq i \leq N \tag{2.20}$$

2. Recursion

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i)\, a_{ij}\right] b_j(\boldsymbol{O}_t), \qquad \begin{array}{l} 1 \leq i \leq N \\ t = 2, 3, \ldots, T \end{array} \tag{2.21}$$

$$\psi_t(j) = \arg\max_i \left[\delta_{t-1}(i)\, a_{ij}\right], \qquad \begin{array}{l} 1 \leq i \leq N \\ t = 2, 3, \ldots, T \end{array} \tag{2.22}$$

3. Termination

$$\hat{P} = \max_i \left[\delta_T(i)\right], \tag{2.23}$$

$$\hat{s_T} = \arg\max_i \left[\delta_T(i)\right]. \tag{2.24}$$

4. Back tracking

$$\hat{s}_t = \psi_{t+1}\left(\hat{s_{t+1}}\right), \quad t = T-1, \ldots, 1. \tag{2.25}$$

It should be noted that the Viterbi algorithm is similar to the forward calculation of Eqs. (2.10)–(2.12). The major difference is the maximization in Eq. (2.21) over previous states, which is used in place of the summation in Eq. (2.11). It also should be clear that a trellis structure efficiently implements the computation of the Viterbi procedure.

2.4 Maximum likelihood estimation of HMM parameters

There is no known method to analytically obtain the model parameter set based on the maximum likelihood (ML) criterion to obtain Λ which maximizes its likelihood $P(\boldsymbol{O} \mid \Lambda)$ for a given observation sequence $\boldsymbol{O}$, in a closed form. Since this problem is a high dimensional nonlinear optimization problem, and there will be a number of local maxima, it is difficult to obtain Λ which globally maximizes $P(\boldsymbol{O} \mid \Lambda)$. However, the model parameter set Λ locally maximizes $P(\boldsymbol{O} \mid \Lambda)$ can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm [27], and the obtained parameter set will be appropriately estimated if a good initial estimate is provided.

In the following, the EM algorithm for the CD-HMM is described. The algorithm for the HMM with discrete output distributions can also be derived in a straightforward manner.

2.4.1 Q-function

In the EM algorithm, an auxiliary function $Q(\Lambda, \hat{\Lambda})$ of the current parameter set Λ and the new parameter set $\hat{\Lambda}$ is defined as follows:

$$Q(\Lambda, \hat{\Lambda}) = \sum_{\text{all } \boldsymbol{S}} P(\boldsymbol{q} \mid \boldsymbol{O}, \Lambda) \log P(\boldsymbol{O}, \boldsymbol{S} \mid \hat{\Lambda}). \tag{2.26}$$

Each mixture of Gaussian components is decomposed into a substate, and $\boldsymbol{S}$ is redefined as a substate sequence,

$$\boldsymbol{S} = \left\{ \left(s_1, m_1\right), \left(s_2, m_2\right), \ldots, \left(s_T, m_T\right) \right\}, \tag{2.27}$$

where (s_t, m_t) represents being in the m_t-th substate (Gaussian component) of the s_t-th state at time t.

At each iteration of the procedure, the current parameter set Λ is replaced by the new parameter set $\hat{\Lambda}$ which maximizes $Q(\Lambda, \hat{\Lambda})$. This iterative procedure can be proved to increase likelihood $P(\boldsymbol{O} \mid \Lambda)$ monotonically and converge to a certain critical point, since it can be proved that the Q-function satisfies the following theorems:

- Theorem 1

$$Q(\Lambda, \hat{\Lambda}) \geq Q(\Lambda, \Lambda) \;\Rightarrow\; P(\boldsymbol{O} \mid \hat{\Lambda}) \geq P(\boldsymbol{O} \mid \Lambda) \tag{2.28}$$

- Theorem 2
  The auxiliary function $Q(\Lambda, \hat{\Lambda})$ has the unique global maximum as a function of Λ, and this maximum is the one and only critical point.

- Theorem 3
  A parameter set Λ is a critical point of the likelihood $P(\boldsymbol{O} \mid \Lambda)$ if and only if it is a critical point of the Q-function.

2.4.2 Maximization of Q-function

According to Eqs. (2.2) and (2.7), $\log P(\boldsymbol{O}, \boldsymbol{S} \mid \Lambda)$ can be written as

$$\log P(\boldsymbol{O}, \boldsymbol{S} \mid \Lambda) = \log P(\boldsymbol{O} \mid \boldsymbol{S}, \Lambda) + \log P(\boldsymbol{S} \mid \Lambda), \tag{2.29}$$

$$\log P(\boldsymbol{O} \mid \boldsymbol{S}, \Lambda) = \sum_{t=1}^{T} \log \mathcal{N}\left(\boldsymbol{O}_t \mid \boldsymbol{\mu}_{s_t w_t}, \boldsymbol{\Sigma}_{s_t w_t}\right), \tag{2.30}$$

$$\log P(\boldsymbol{S} \mid \Lambda) = \log \pi_{q_1} + \sum_{t=2}^{T} \log a_{s_{t-1} s_t} + \sum_{t=1}^{T} \log w_{s_t s_t}. \tag{2.31}$$

Hence, Q-function (Eq. (2.26)) can be rewritten as

$$
\begin{aligned}
Q(\Lambda, \hat{\Lambda}) = & \sum_{i=1}^{N} P\left(\boldsymbol{O}, s_1 = i \mid \Lambda\right) \cdot \log \pi_i \\
& + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P\left(\boldsymbol{O}, s_t = i, s_{t+1} = j\right) \cdot \log a_{ij} \\
& + \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} P\left(\boldsymbol{O}, s_t = i, m_t = m \mid \Lambda\right) \cdot \log w_{im} \\
& + \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} P\left(\boldsymbol{O}, s_t = i, m_t = m \mid \Lambda\right) \cdot \log \mathcal{N}\left(\boldsymbol{O}_t \mid \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}\right). \qquad (2.32)
\end{aligned}
$$

The parameter set Λ which maximizes the above equation subject to the stochastic constraints

$$\sum_{i=1}^{N} \pi_i = 1, \qquad (2.33)$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \leq i \leq N \qquad (2.34)$$

$$\sum_{m=1}^{M} w_{im} = 1, \quad 1 \leq i \leq N \qquad (2.35)$$

can be derived by Lagrange multipliers or differential calculus as follows [28]:

$$\pi_i = \gamma_1(i), \qquad 1 \le i \le N \tag{2.36}$$

$$a_{ij} = \frac{\sum_{t=2}^{T} \xi_{t-1}(i, j)}{\sum_{t=2}^{T} \gamma_{t-1}(i)}, \qquad \begin{array}{l} 1 \le i \le N \\ 1 \le j \le N \end{array} \tag{2.37}$$

$$w_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m)}{\sum_{t=1}^{T} \gamma_t(i)}, \qquad \begin{array}{l} 1 \le i \le N \\ 1 \le m \le M \end{array} \tag{2.38}$$

$$\boldsymbol{\mu}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m) \cdot \boldsymbol{O}_t}{\sum_{t=1}^{T} \gamma_t(i, m)}, \qquad \begin{array}{l} 1 \le i \le N \\ 1 \le m \le M \end{array} \tag{2.39}$$

$$\boldsymbol{\Sigma}_{im} = \frac{\sum_{t=1}^{T} \gamma_t(i, m) \cdot (\boldsymbol{O}_t - \boldsymbol{\mu}_{im}) (\boldsymbol{O}_t - \boldsymbol{\mu}_{im})^\top}{\sum_{t=1}^{T} \gamma_t(i, m)}, \qquad \begin{array}{l} 1 \le i \le N \\ 1 \le m \le M \end{array} \tag{2.40}$$

where $\gamma_t(i)$, $\gamma_t(i, m)$, and $\xi_t\,(i, j)$ are the probability of being in the j-th state at time t, the probability of being in the m-th substate of the i-th state at time t, and the probability of being in the i-th state at time t and j-th state at time $t + 1$, respectively, that is

$$\begin{aligned}\gamma_t(i) &= P(\boldsymbol{O}, q_t = i \mid \Lambda) \\ &= \frac{\alpha_t(i)\beta(i)}{\sum\limits_{j=1}^{N} \alpha_t(j)\beta_t(j)}, \qquad \begin{matrix} 1 \le i \le N \\ t = 1, \ldots, T \end{matrix} \end{aligned} \tag{2.41}$$

$$\begin{aligned}\gamma_t(i,m) &= P(\boldsymbol{O}, q_t = i, s_t = m \mid \Lambda) \\ &= \frac{\alpha_t(i)\beta(i)}{\sum\limits_{j=1}^{N} \alpha_t(j)\beta_t(j)} \cdot \frac{w_{im}\mathcal{N}(\boldsymbol{O}_t \mid \boldsymbol{\mu}_{im}, \boldsymbol{\sigma}_{im})}{\sum\limits_{k=1}^{M} w_{ik}\mathcal{N}(\boldsymbol{O}_t \mid \boldsymbol{\mu}_{ik}, \boldsymbol{\sigma}_{ik})}, \quad \begin{matrix} 1 \le i \le N \\ 1 \le m \le M \\ t = 1, \ldots, T \end{matrix} \end{aligned} \tag{2.42}$$

$$\begin{aligned}\xi_t(i,j) &= P(\boldsymbol{O}, q_t = i, q_{t+1} = j \mid \Lambda) \\ &= \frac{\alpha_t(i)a_{ij}b_j(\boldsymbol{O}_{t+1})\beta_{t+1}(j)}{\sum\limits_{l=1}^{N}\sum\limits_{n=1}^{N} \alpha_t(l)a_{ln}b_n(\boldsymbol{O}_{t+1})\beta_{t+1}(n)}. \qquad \begin{matrix} 1 \le i \le N \\ t = 1, \ldots, T \end{matrix} \end{aligned} \tag{2.43}$$

2.5 Summary

In this chapter, the basic theories of the hidden Markov models (HMMs), its algorithm for calculating the output probability (forward-backward algorithm), searching the optimal state sequence (Viterbi algorithm), and estimating its parameters (EM algorithm) are described. Following chapters show the separable lattice 2-D HMMs, one of the HMM based approach for image recognition.

Chapter 3

Separable Lattice 2-D Hidden Markov Models

This chapter describes separable lattice 2-D HMMs (SL2D-HMMs). SL2D-HMMs have the composite structure of multiple hidden state sequences which interact to model the observation on a lattice. SL2D-HMMs perform an elastic matching in both horizontal and vertical directions; this makes it possible to model not only invariance to the size and location of an object but also nonlinear warping in each dimension.

The rest of this chapter is organized as follows. Section 3.1 reviews the previous works of HMM-based image recognition techniques. Section 3.2 describes the model definition of SL2D-HMMs. Section 3.3 derives the training algorithms of SL2D-HMMs.

3.1 Related work

Statistical approaches have been successfully applied in the field of image recognition. In particular, principal component analysis (PCA) based approaches, such as the eigenface (eigen-image) method [29] and subspace method [30], attain good recognition performance. There are many significant classifiers and feature representations. However, in the case of conventional methods, some pre-processing for normalizing image variations, e.g., geometric variations such as size, location, and rotation, is usually applied to input images because many classifiers cannot deal with such image variations. The accuracy of these normalization processes affects recognition performance. Task-dependent normalization techniques have thus been developed for each image recognition task. However, the final objective of image recognition is not to accurately normalize image variations for human perception but to achieve high recognition performance. It is therefore a good idea to integrate the normalization processes into classifiers and optimize them on the basis of a consistent criterion.

HMM based techniques for image recognition have been proposed to reduce the influence of geometric variations [3–13]. 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 [3]. 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.

In previous work [31], planar hidden Markov models were developed to provide a probabilistic formulation for the planar warping problem. The probability of a particular state depends only on the state at adjacent observations in both horizontal and vertical directions. This assumption is a natural extension of the Markov property to a second-order source and the complexity can be reduced by generalizing the optimality principle as in the one-dimensional forward-backward and Viterbi algorithms. However, the computation of planar HMMs is still exponential. Therefore, approximate Viterbi training algorithms (e.g. [32]) and additional assumptions to simplify the model structure (e.g. [6]) have been proposed to solve the problem in polynomial time.

On the other hand, a more restricted structure, pseudo 2-D HMMs (or called embedded HMMs) have been proposed [4] and applied to many image recognition tasks. Their extension to pseudo 3-D HMMs has also been developed for image sequence recognition [33]. A pseudo 2-D HMM has a composite state structure for an efficient 2-D representation avoiding the complexity burden of a fully connected 2-D HMM. Figure 3.1 shows the graphical model representation of the pseudo 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 reduces the computational complexity and the maximum likelihood training algorithm has been derived [5]. However, the state alignments of consecutive observation lines in the vertical direction are calculated independently of each other and this hypothesis 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

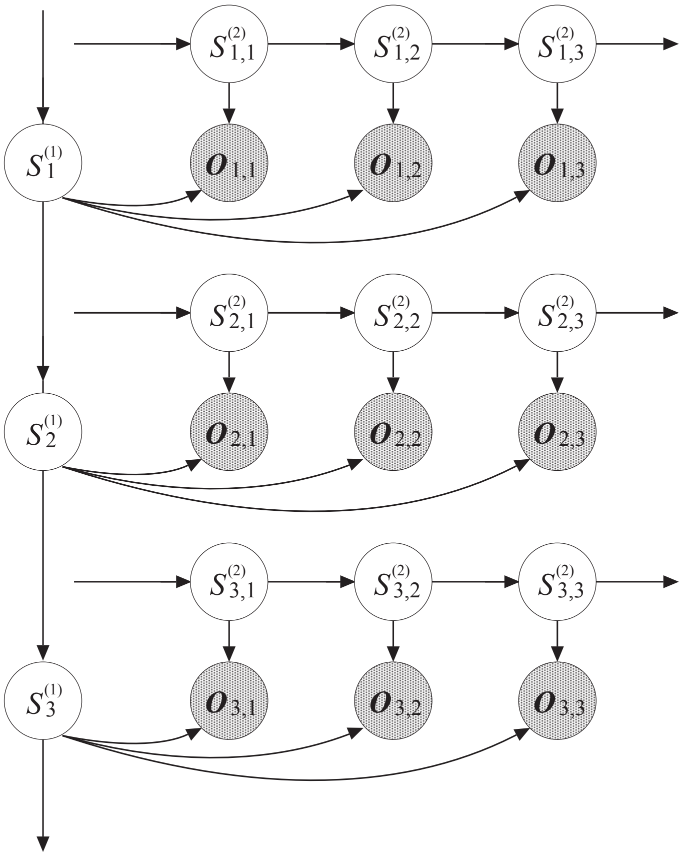


Figure 3.1: Graphical model representation of the pseudo 2-D HMMs: The states of a superior HMM in the horizontal direction are called super-states and each super-state has a one-dimensional HMM in the vertical direction.

some efficient approximation algorithms have been proposed for the 2D-DP problem [14–17], 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) have been proposed [9] to reduce the computational complexity while retaining the good properties for modeling multi-dimensional data. The detail of SL2D-HMMs will be described in the next section.

3.2 Model definition

Separable lattice 2-D hidden Markov models (SL2D-HMMs) [9] are defined for modeling two-dimensional data. The observations of two-dimensional data, e.g., pixel values of an image and image sequence, are assumed to be given on a two-dimensional lattice:

$$\boldsymbol{O} = \{\boldsymbol{O}_{\boldsymbol{t}} | \boldsymbol{t} = (t^{(1)}, t^{(2)}) \in \boldsymbol{T}\}, \tag{3.1}$$

where $\boldsymbol{t}$ denotes the coordinates of the lattice in two dimensional space $\boldsymbol{T}$ and $t^{(m)} = 1, \ldots, T^{(m)}$ is the coordinate of the m-th dimension. The observation $\boldsymbol{O}_{\boldsymbol{t}}$ is emitted from the state indicated by the hidden variable $\boldsymbol{S}_{\boldsymbol{t}} \in \boldsymbol{K}$. The hidden variables $\boldsymbol{S}_{\boldsymbol{t}} \in \boldsymbol{K}$ can take one of $K = K^{(1)}K^{(2)}$ states, which are assumed to be arranged on a two-dimensional state lattice $\boldsymbol{K} = \{(1,1), (1,2), \ldots, (1, K^{(2)}), (2,1), \ldots, (K^{(1)}, K^{(2)})\}$. In other words, a set of hidden variables, $\{\boldsymbol{S}_{\boldsymbol{t}} | \boldsymbol{t} \in \boldsymbol{T}\}$, 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 $\boldsymbol{O}_{\boldsymbol{t}}$ is dependent only on the state $\boldsymbol{S}_{\boldsymbol{t}}$ as in ordinary HMMs, dependencies among hidden variables determine the properties and the modeling ability of two-dimensional HMMs.

To reduce the number of possible state sequences, we constrain the hidden variables to be composed of two Markov chains:

$$\begin{aligned} \boldsymbol{S} &= \{\boldsymbol{S}^{(1)}, \boldsymbol{S}^{(2)}\}, & (3.2)\\ \boldsymbol{S}^{(m)} &= \{S^{(m)}_{1}, \ldots, S^{(m)}_{t^{(m)}}, \ldots, S^{(m)}_{T^{(m)}}\}, & (3.3) \end{aligned}$$

where $\boldsymbol{S}^{(m)}$ is the Markov chain along with the m-th coordinate and $S^{(m)}_{t^{(m)}} \in \{1, \ldots, K^{(m)}\}$. In the separable lattice 2-D HMMs, the composite structure of hidden variables is defined as the product of hidden state sequences: $\boldsymbol{S}_{\boldsymbol{t}} = (S^{(1)}_{t^{(1)}}, S^{(2)}_{t^{(2)}})$. 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 $\prod_m \{K^{(m)}\}^{T^{(m)}}$.

The joint probability of observation vectors $\boldsymbol{O}$ and hidden variables $\boldsymbol{S}$ can be written as

$$\begin{aligned} P(\boldsymbol{O}, \boldsymbol{S} \mid \Lambda) &= P(\boldsymbol{O} \mid \boldsymbol{S}, \Lambda) \prod_{m=1,2} P(\boldsymbol{S}^{(m)} \mid \Lambda) \\ &= \prod_{\boldsymbol{t}} P(\boldsymbol{O}_{\boldsymbol{t}} \mid \boldsymbol{S}_{\boldsymbol{t}}, \Lambda) \prod_{m=1,2} \left[P(S^{(m)}_{1} \mid \Lambda) \prod_{t^{(m)}=2}^{T^{(m)}} P(S^{(m)}_{t^{(m)}} \mid S^{(m)}_{t^{(m)}-1}, \Lambda) \right] \end{aligned} \tag{3.4}$$

where Λ is the model parameters of SL2D-HMMs. This model parameters of SL2D-HMMs are summarized as follows:

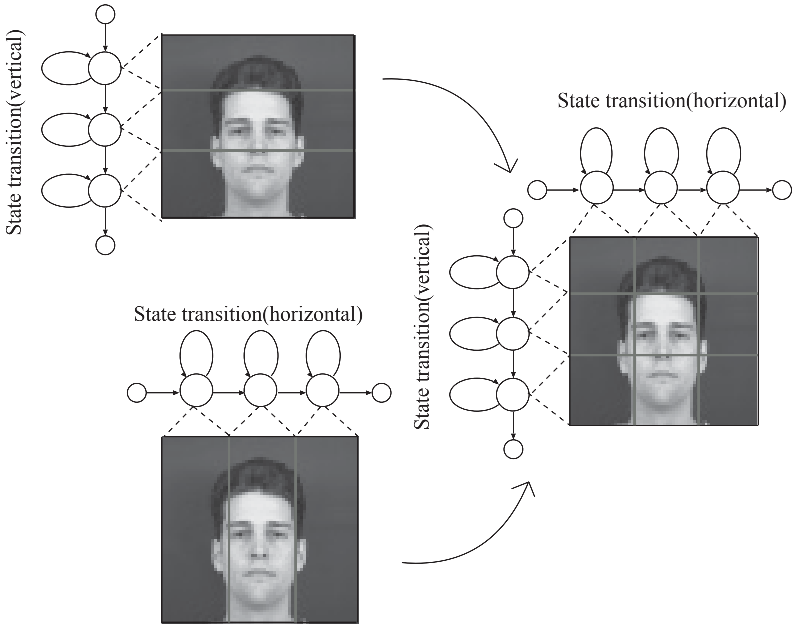

State transition(vertical)
State transition(horizontal)
State transition(horizontal)
State transition(vertical)

Figure 3.2: Model structure of the separable lattice 2-D HMMs: hidden state sequences are composed of independent two Markov chains.

- **Parameters for state transition probability**:

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

  $$\pi_i^{(m)} = P(S_1^{(m)} = i \mid \Lambda) \tag{3.5}$$

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

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

  $$a_{ij}^{(m)} = P(S_{t^{(m)}}^{(m)} = j \mid S_{t^{(m)}-1}^{(m)} = i, \Lambda) \tag{3.6}$$

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

- **Parameters for output probability distribution** :
  $\boldsymbol{B} = \{b_{\boldsymbol{k}}(\boldsymbol{O_t}) | \boldsymbol{k} \in \boldsymbol{K}\}$: the output probability distributions, where $b_{\boldsymbol{k}}(\boldsymbol{O_t})$ is the probability of observation vector $\boldsymbol{O_t}$ at the state $\boldsymbol{k}$ on the state lattice $\boldsymbol{K}$ and assumed to be a single Gaussian distribution :

  $$P(\boldsymbol{O_t} \mid \boldsymbol{S_t} = \boldsymbol{k}) = \mathcal{N}(\boldsymbol{O_t}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) \tag{3.7}$$

  where $\boldsymbol{\mu_k}$ and $\boldsymbol{\Sigma_k}$ denote the "state level" mean vector and the covariance matrix, respectively.

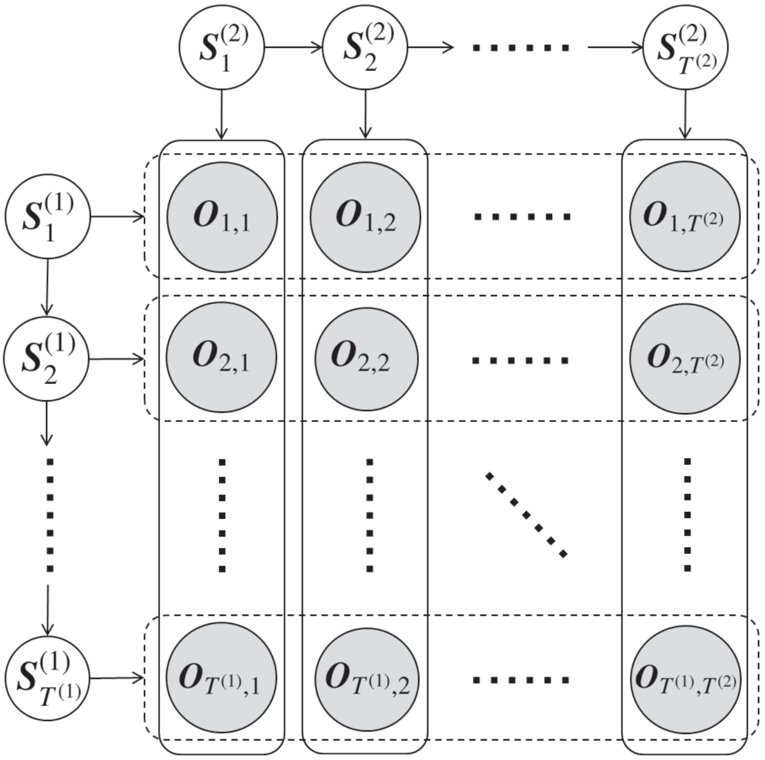


Figure 3.3: Graphical model representation of the separable lattice 2-D HMMs: The rounded box represents a group of variables and the arrow to the box represents the dependency to all variables in the box instead of drawing arrows to the all variables. The observations are emitted from the product of horizontal and vertical hidden state sequences.

Using the above shorthand notation, a separable lattice 2-D HMM is defined as

$$\Lambda = \{\Lambda^{(1)}, \Lambda^{(2)}\boldsymbol{B}\}, \tag{3.8}$$

$$\Lambda^{(m)} = \{\Pi^{(m)}, \boldsymbol{A}^{(m)}\}. \tag{3.9}$$

Fig. 3.2 and 3.3 show the model structure of the separable lattice 2-D HMMs and its graphical model representation, respectively. In Fig. 3.3, 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 separable 2-D lattice HMMs can be applied to image modeling and perform an elastic matching in both horizontal and vertical directions by assuming the transition probabilities with left-to-right and top-to-bottom topologies. Although the structure of the proposed model cannot represent rotations of images, it is still useful for image detection and the framework makes it possible to achieve size- and location-invariant image recognition. Furthermore, the proposed model can be used for 3-D and higher dimensional applications, e.g., image sequences, 3-D object models, etc., due to the composite structure which reduces the complexity of the algorithm while retaining the good properties for modeling multi-dimensional data.

The main difference between the proposed model and the embedded HMMs is that the SL2D-HMM have a symmetric structure in vertical and horizontal directions. Therefore, there is no need to determine which direction of 2-D data should be modeled as the super states or the embedded states. If the hidden variables of the embedded states also shared for all observation sequences, an embedded HMM becomes equivalent to a SL2D-HMM. In the embedded HMMs, the exact EM algorithm can be performed in practice, because the state transitions of an embedded state sequence depend only on the corresponding super state. However, in SL2D-HMMs, the state transitions of one direction depend on the all the hidden variables of the other direction; therefore the exact EM algorithm becomes infeasible.

In the next section, the training algorithm for the SL2D-HMMs using the variational EM algorithm and the variational DAEM algorithm and are derived. Although some extensions of SL2D-HMMs have been proposed, e.g., explicit state duration modeling [12], this dissertation uses an original form of SL2D-HMMs.

3.3 Training algorithm

3.3.1 EM algorithm

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

$$Q(\Lambda, \Lambda') = \sum_{\boldsymbol{S}} P(\boldsymbol{S} \mid \boldsymbol{O}, \Lambda) \ln P(\boldsymbol{O}, \boldsymbol{S} \mid \Lambda') \tag{3.10}$$

The likelihood of the training data is guaranteed to increase by increasing the value of the Q-function:

$$Q(\Lambda, \Lambda') \geq Q(\Lambda, \Lambda) \;\Rightarrow\; P(\boldsymbol{O} \mid \Lambda') \geq P(\boldsymbol{O} \mid \Lambda) \tag{3.11}$$

The EM algorithm starts with some initial model parameters and iterates between the following two steps:

$$\begin{aligned}\text{(E-step)} &\quad : \quad \text{compute } Q(\Lambda^{(k)}, \Lambda) \\ \text{(M-step)} &\quad : \quad \Lambda^{(k+1)} = \arg\max_{\Lambda} Q(\Lambda^{(k)}, \Lambda)\end{aligned}$$

where k denotes the iteration number. The E-step computes the posterior probabilities over the hidden states while keeping model parameters fixed to current values. The M-step uses these probabilities to calculate the expected log-likelihood of the training data as a function of the parameters and maximize the Q-function with respect to model parameters. In this procedure, each step increases the value of the Q-function; hence the likelihood of the training data is also guaranteed to increase or remain unchanged on each iteration.

By maximizing the Q-function with respect to model parameters Λ, the re-estimation formula in the M-step can be easily derived as follows:

$$\pi_i^{(m)} = \left\langle S_1^{(m)}, i \right\rangle \tag{3.12}$$

$$a_{ij}^{(m)} = \frac{\sum_{t^{(m)}=2}^{T^{(m)}} \left\langle \left(S_{t^{(m)}-1}^{(m)}, i\right) \left(S_{t^{(m)}}^{(m)}, j\right) \right\rangle}{\sum_{t^{(m)}=2}^{T^{(m)}} \left\langle S_{t^{(m)}}^{(m)}, i \right\rangle} \tag{3.13}$$

$$\boldsymbol{\mu_k} = \frac{\sum_{\boldsymbol{t}} \langle \boldsymbol{S_t}, \boldsymbol{k} \rangle \boldsymbol{O_t}}{\sum_{\boldsymbol{t}} \langle \boldsymbol{S_t}, \boldsymbol{k} \rangle} \tag{3.14}$$

$$\boldsymbol{\Sigma_k} = \frac{\sum_{\boldsymbol{t}} \langle \boldsymbol{S_t}, \boldsymbol{k} \rangle (\boldsymbol{O_t} - \boldsymbol{\mu_k})(\boldsymbol{O_t} - \boldsymbol{\mu_k})^\top}{\sum_{\boldsymbol{t}} \langle \boldsymbol{S_t}, \boldsymbol{k} \rangle} \tag{3.15}$$

where $\langle \cdot \rangle$ denotes the expectation with respect to the posterior distribution over the hidden variables. These expectations are computed in the E-step by the following equations.

$$\left\langle \left(S_{t^{(m)}}^{(m)}, i\right) \right\rangle = \sum_{\boldsymbol{S}} P(\boldsymbol{S} \mid \boldsymbol{O}, \Lambda) \delta(S_{t^{(m)}}^{(m)}, i) \tag{3.16}$$

$$\left\langle \left(S_{t^{(m)}-1}^{(m)}, i\right) \left(S_{t^{(m)}}^{(m)}, j\right) \right\rangle = \sum_{\boldsymbol{S}} P(\boldsymbol{S} \mid \boldsymbol{O}, \Lambda) \delta(S_{t^{(m)}-1}^{(m)}, i)(S_{t^{(m)}}^{(m)}, j) \tag{3.17}$$

$$\langle \boldsymbol{S_t}, \boldsymbol{k} \rangle = \left\langle S_{t^{(1)}}^{(1)}, k^{(1)} \right\rangle \left\langle S_{t^{(2)}}^{(2)}, k^{(2)} \right\rangle \tag{3.18}$$

In the case of a 1-D HMM, the forward-backward algorithm is applied to calculate the expectations efficiently. Even though the Markov chains of SL2D- are independent a-priori, they become conditionally dependent given the observations and the computation of expectations become infeasible. If we compute expectations in the exact E-step directly according to Eqs.(3.16)–(3.18), we need to consider summations over all the combinations of states and the complexity of the E-step is $O\left(\Pi_m \{K^{(m)}\}^{T^{(m)}}\right)$. As in one-dimensional HMMs, the complexity of SL2D-HMMs can be reduced by using forward-backward algorithm. The $\boldsymbol{Q}$-function can be rewritten with respect to one of Markov chains $\boldsymbol{S}^{(n)}$ as follows:

$$
\begin{aligned}
&\sum_{\boldsymbol{S}} P(\boldsymbol{S} \mid \boldsymbol{O}, \Lambda) \ln P(\boldsymbol{S}, \boldsymbol{O} \mid \Lambda') \\
&\quad = \sum_{\overline{\boldsymbol{S}} \in \boldsymbol{S} \backslash \boldsymbol{S}^{(n)}} \sum_{\boldsymbol{S}^{(n)}} P(\boldsymbol{S}^{(n)} \mid \overline{\boldsymbol{S}}, \boldsymbol{O}, \Lambda) P(\overline{\boldsymbol{S}} \mid \boldsymbol{O}, \Lambda) \ln P(\boldsymbol{S}^{(n)}, \boldsymbol{O} \mid \overline{\boldsymbol{S}}, \Lambda) \\
&\quad = \sum_{\overline{\boldsymbol{S}} \in \boldsymbol{S} \backslash \boldsymbol{S}^{(n)}} P(\overline{\boldsymbol{S}} \mid \boldsymbol{O}, \Lambda) \times \left[\sum_{\boldsymbol{S}^{(n)}} P(\boldsymbol{S}^{(n)} \mid \overline{\boldsymbol{S}}, \boldsymbol{O}, \Lambda) \ln P(\boldsymbol{S}^{(n)}, \boldsymbol{O} \mid \overline{\boldsymbol{S}}, \Lambda) \right] \\
&\qquad + \sum_{\overline{\boldsymbol{S}} \in \boldsymbol{S} \backslash \boldsymbol{S}^{(n)}} P(\overline{\boldsymbol{S}} \mid \boldsymbol{O}, \Lambda) \ln P(\boldsymbol{S} \mid \Lambda)
\end{aligned}
\tag{3.19}
$$

where the term in the square bracket is the $\boldsymbol{Q}$-function associated with $\boldsymbol{S}^{(n)}$ given $\overline{\boldsymbol{S}}$ and that can be calculated by the forward-backward algorithm. Hence the complexity of the exact E-step can be reduced to $O(\{K^{(n)}\}^2 T^{(n)} \Pi_{m+n} \{K^{(m)}\}^{T^{(m)}})$. However, the calculation of the posterior distribution $P(\boldsymbol{S} \mid \boldsymbol{O}, \Lambda)$ in the E-step is computationally intractable due to the combination of hidden variables. To derive a feasible problem, we applied the variational EM algorithm [18] to the training algorithm of SL2D-HMMs.

3.3.2 Variational EM algorithm

The variational methods approximate the posterior distribution over the hidden variables by a tractable distribution. Any distribution over the hidden variables defines a lower bound on the log-likelihood

$$
\begin{aligned}
\ln P(\boldsymbol{O} \mid \Lambda) &= \ln \sum_{\boldsymbol{S}} Q(\boldsymbol{S}) \frac{P(\boldsymbol{O}, \boldsymbol{S} \mid \Lambda)}{Q(\boldsymbol{S})} \\
&\geq \sum_{\boldsymbol{S}} Q(\boldsymbol{S}) \ln \frac{P(\boldsymbol{O}, \boldsymbol{S} \mid \Lambda)}{Q(\boldsymbol{S})} \\
&= \mathcal{F}(Q, \Lambda)
\end{aligned}
\tag{3.20}
$$

where Jensen's inequality has been applied. The difference between $\ln P(\boldsymbol{O} \mid \Lambda)$ and $\mathcal{F}$ is given by the KL divergence between $Q(\boldsymbol{S})$ and the posterior distribution of the hidden variables $P(\boldsymbol{S} \mid \boldsymbol{O}, \Lambda)$:

$$\begin{aligned}
\mathcal{F}(Q, \Lambda) &= \sum_{\boldsymbol{S}} Q(\boldsymbol{S}) \ln \frac{P(\boldsymbol{O}, \boldsymbol{S} \mid \Lambda)}{Q(\boldsymbol{S})} \\
&= \sum_{\boldsymbol{S}} Q(\boldsymbol{S}|\Lambda) \ln P(\boldsymbol{O} \mid \Lambda) + \sum_{\boldsymbol{S}} Q(\boldsymbol{S}) \ln \frac{P(\boldsymbol{S} \mid \boldsymbol{O}, \Lambda)}{Q(\boldsymbol{S})} \\
&= \ln P(\boldsymbol{O} \mid \Lambda) - \mathrm{KL}(Q \,\|\, P) \qquad (3.21)
\end{aligned}$$

Since the true log-likelihood $\ln P(\boldsymbol{O} \mid \Lambda)$ is independent of $Q(\boldsymbol{S})$, maximizing the lower bound $\mathcal{F}$ is equivalent to minimizing the KL divergence. If we allow $Q(\boldsymbol{S})$ to have complete flexibility then we see that the optimal $Q(\boldsymbol{S})$ distribution is given by the true posterior $P(\boldsymbol{S} \mid \boldsymbol{O}, \Lambda)$, in the case where the KL divergence is zero and the bound becomes exact. In order to yield a tractable algorithm, it is necessary to consider a more restricted structure of $Q(\boldsymbol{S})$ distributions. Given the structure, the parameters of $Q(\boldsymbol{S})$ are varied so as to obtain the tightest possible bound, which maximizes $\mathcal{F}$.

The variational EM algorithm iteratively maximizes $\mathcal{F}$ with respect to the Q and Λ holding the other parameters fixed:

$$\begin{aligned}
\text{(E-step)} \quad &: \quad Q^{(k+1)} = \arg\max_{Q \in C} \mathcal{F}(Q, \Lambda^{(k)}) \\
\text{(M-step)} \quad &: \quad \Lambda^{(k+1)} = \arg\max_{\Lambda} \mathcal{F}(Q^{(k+1)}, \Lambda)
\end{aligned}$$

where C is the set of constrained distributions. In this procedure, the lower bound $\mathcal{F}$ is guaranteed to increase instead of the value of the $\mathcal{Q}$-function.

The complexity and the approximation property of the variational EM algorithm are dependent on a constraint to the posterior distribution $Q(\boldsymbol{S})$ and it should be determined for each structure of graphical models. Here we consider a constrained family of variational distributions for the proposed model by assuming that $Q(\boldsymbol{S})$ factorizes over subset $\boldsymbol{S}^{(m)}$ of the variables in $\boldsymbol{S}$, so that

$$Q(\boldsymbol{S}) = Q(\boldsymbol{S}^{(1)})Q(\boldsymbol{S}^{(2)}) \qquad (3.22)$$

where $Q(\boldsymbol{S})$ are the posterior distribution over $\boldsymbol{S}$ and $\sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) = 1$, $m = 1, 2$. To make the bound as tight as possible, we use elementary calculus of variations to take functional derivatives of the lower bound with respect to $Q(\boldsymbol{S}^{(m)})$. In this case, the Euler-Lagrange equation can be solved simply by taking partial derivatives with respect to one of the distributions:

$$
\begin{aligned}
&\frac{\partial \mathcal{F}}{\partial Q\left(\boldsymbol{S}^{(n)} = \boldsymbol{S}^{(n)'}\right)} \\
&= \sum_{\overline{\boldsymbol{S}} \in \boldsymbol{S} \backslash \boldsymbol{S}^{(n)}} \prod_{m \neq n} Q(\boldsymbol{S}^{(m)}) \ln P(\boldsymbol{O}, \overline{\boldsymbol{S}}, \boldsymbol{S}^{(n)'} \mid \Lambda) - \ln Q(\boldsymbol{S}^{(n)'}) - 1 \\
&= \sum_{\overline{\boldsymbol{S}} \in \boldsymbol{S} \backslash \boldsymbol{S}^{(n)}} \prod_{m \neq n} Q(\boldsymbol{S}^{(m)}) \ln P(\boldsymbol{O} \mid \overline{\boldsymbol{S}}, \boldsymbol{S}^{(n)'}, \Lambda) + \ln P(\boldsymbol{S}^{(n)'}) - \ln Q(\boldsymbol{S}^{(n)'}) + const. \quad (3.23)
\end{aligned}
$$

The maximum of $\mathcal{F}$ occurs at a critical point subject to the constraint that $\sum_{\boldsymbol{S}^{(n)}} Q(\boldsymbol{S}^{(n)}) = 1$, and can be found using a Lagrange multiplier $\lambda^{(n)}$. By setting for each of state sequence $\boldsymbol{S}^{(n)}$, following Euler-Lagrange equation can be obtained.

$$\frac{\partial \mathcal{F}}{\partial Q(^{(n)})} + \lambda^{(n)} = 0. \quad (3.24)$$

The optimal distributions can be derived as

$$Q(\boldsymbol{S}^{(n)}) = \frac{1}{Z^{(n)}} P(\boldsymbol{S}^{(n)}|\Lambda) \prod_{t^{(n)}=1}^{T^{(n)}} h(t^{(n)}, S^{(n)}_{t^{(n)}}) \quad (3.25)$$

$$
\begin{aligned}
\ln h(t^{(n)}, S^{(n)}_{t^{(n)}}) &= \sum_{\bar{\boldsymbol{S}} \in \boldsymbol{S} \backslash \boldsymbol{S}^{(n)}} \prod_{m \neq n} Q(\bar{\boldsymbol{S}}^{(m)}) \sum_{\bar{\boldsymbol{t}} \in \boldsymbol{t} \backslash t^{(n)}} \ln P(\boldsymbol{O}_{\bar{\boldsymbol{t}}, t^{(n)}} | \bar{\boldsymbol{S}}, S^{(n)}_{t^{(n)}}, \Lambda) \quad (3.26) \\
&= \sum_{\bar{\boldsymbol{k}} \in \boldsymbol{k} \backslash k^{(n)}} \sum_{\bar{\boldsymbol{t}} \in \boldsymbol{t} \backslash t^{(n)}} \prod_{m \neq n} \langle (S^{(m)}_{\bar{t}^{(m)}}, \bar{k}^{(m)}) \rangle \ln P(\boldsymbol{O}_{\bar{\boldsymbol{t}}, t^{(n)}} | \boldsymbol{S}_{\bar{\boldsymbol{t}}} = \bar{\boldsymbol{k}}, S^{(n)}_{t^{(n)}}, \Lambda) \quad (3.27)
\end{aligned}
$$

where $Z^{(n)}$ is a normalization constant including $\lambda^{(n)}$. By inspection, this distribution is the same structure as the posterior of standard HMMs: the expectation $h(t^{(n)}, S^{(n)}_{t^{(n)}})$ corresponds to the observation probability associated with the state variable $S^{(n)}_{t^{(n)}}$. Therefore, the forward-backward algorithm can be used to compute the following expectations efficiently:

$$
\begin{aligned}
\left\langle \left(S^{(m)}_{t^{(m)}}, i\right) \right\rangle &= \sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) \delta(S^{(m)}_{t^{(m)}}, i) \quad (3.28) \\
\left\langle \left(S^{(m)}_{t^{(m)}-1}, i\right) \left(S^{(m)}_{t^{(m)}}, j\right) \right\rangle &= \sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) \delta(S^{(m)}_{t^{(m)}-1}, i) \delta(S^{(m)}_{t^{(m)}}, j) \quad (3.29) \\
\langle (\boldsymbol{S}_t, \boldsymbol{k}) \rangle &= \prod_{m=1}^{M} \langle (S^{(m)}_{t^{(m)}}, k^{(m)}) \rangle \quad (3.30)
\end{aligned}
$$

The complexity of E-step with the variational approximation becomes $O(M \prod_m K^{(m)} T^{(m)})$ owing to the computation of $\ln h(t^{(n)}, S^{(n)}_{t^{(n)}})$. Note that the computational cost can be significantly reduced from the exact EM algorithm to polynomial time complexity.

Using these expectations, the re-estimation formula of the proposed model in the M-step are derived as follows:

$$\pi_i^{(m)} = \left\langle \left(S_1^{(m)}, i \right) \right\rangle, \tag{3.31}$$

$$a_{ij}^{(m)} = \frac{\sum_{t^{(m)}=2}^{T^{(m)}} \left\langle \left(S_{t^{(m)}-1}^{(m)}, i \right) \left(S_{t^{(m)}}^{(m)}, j \right) \right\rangle}{\sum_{t^{(m)}=1}^{T^{(m)}} \left\langle \left(S_{t^{(m)}}^{(m)}, i \right) \right\rangle}, \tag{3.32}$$

$$\boldsymbol{\mu}_{\boldsymbol{k}} = \frac{\sum_{\boldsymbol{t}} \langle \boldsymbol{S}_{\boldsymbol{t}}, \boldsymbol{k} \rangle \, \boldsymbol{O}_{\boldsymbol{t}}}{\sum_{\boldsymbol{t}} \langle \boldsymbol{S}_{\boldsymbol{t}}, \boldsymbol{k} \rangle} \tag{3.33}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{k}} = \frac{\sum_{\boldsymbol{t}} \langle \boldsymbol{S}_{\boldsymbol{t}}, \boldsymbol{k} \rangle \, (\boldsymbol{O}_{\boldsymbol{t}} - \boldsymbol{\mu}_{\boldsymbol{k}})(\boldsymbol{O}_{\boldsymbol{t}} - \boldsymbol{\mu}_{\boldsymbol{k}})^{\top}}{\sum_{\boldsymbol{t}} \langle \boldsymbol{S}_{\boldsymbol{t}}, \boldsymbol{k} \rangle} \tag{3.34}$$

3.3.3 Variational DAEM algorithm

The EM algorithm has the problem that the solution converges to a local optimum and the convergence point depends on the initial model parameters. In the variational EM algorithm for SL2D-, the decoupled posterior distributions are updated individually based not only on the initial model parameters but also on the other distributions, both of which are unreliable at an early stage of training. To avoid this problem, the deterministic annealing EM (DAEM) algorithm [34] can be applied to the algorithm derived in the previous section and it is shown that the expectations with respect to the decoupled posterior distributions for the DAEM can also be calculated by the forward-backward procedure.

In the DAEM algorithm, the problem of maximizing the log-likelihood is reformulated as minimizing the thermodynamic free energy defined as

$$\mathcal{L}_\beta = -\frac{1}{\beta} \ln \sum_{\boldsymbol{S}} P(\boldsymbol{S}, \boldsymbol{O} | \Lambda)^\beta \tag{3.35}$$

where $1/\beta$ called the "temperature" and this cost function can be rewritten by using

Jensen's inequality:

$$\begin{aligned}
-\mathcal{L}_\beta &= \frac{1}{\beta}\ln\sum_{S} Q_\beta(\boldsymbol{S})\frac{P(\boldsymbol{S},\boldsymbol{O}|\Lambda)^\beta}{Q_\beta(\boldsymbol{S})} \\
&\geq \frac{1}{\beta}\sum_{S} Q_\beta(\boldsymbol{S})\ln\frac{P(\boldsymbol{S},\boldsymbol{O}|\Lambda)^\beta}{Q_\beta(\boldsymbol{S})} \\
&= \sum_{S} Q_\beta(\boldsymbol{S})\ln P(\boldsymbol{S},\boldsymbol{O}|\Lambda) - \frac{1}{\beta}\sum_{S} Q_\beta(\boldsymbol{S})\ln Q_\beta(\boldsymbol{S}) \qquad (3.36) \\
&= \mathcal{F}_\beta(Q_\beta,\Lambda) \qquad (3.37)
\end{aligned}$$

where $-\mathcal{F}_\beta(Q_\beta,\Lambda)$ is the same form as the free energy in statistical physics, and maximizing $\mathcal{F}_\beta(Q_\beta,\Lambda)$ with a fixed temperature can be interpreted as the approach to thermodynamic equilibrium. In the algorithm, the temperature is gradually decreased and the function is deterministically optimized at each temperature. The procedure of the DAEM algorithm can be summarized as follows:

1 Give an initial model and set $\beta = \beta_{min}$

2 Iterate EM-steps with β fixed until F_β converged:

$$\begin{aligned}
\text{(E step)} &: \quad Q_\beta^{(k+1)} = \underset{Q_\beta \in C}{\operatorname{argmax}} \mathcal{F}_\beta(Q_\beta,\Lambda^{(k)}) \\
\text{(M step)} &: \quad \Lambda^{(k+1)} = \underset{\Lambda}{\operatorname{argmax}} \mathcal{F}_\beta(Q_\beta^{(k+1)},\Lambda)
\end{aligned}$$

3 Increase β.

4 If $\beta > 1$, stop the procedure. Otherwise go to step 2.

where $1/\beta_{min}$ is an initial temperature and should be chosen as a high enough value that the EM-steps can achieve a single global maximum of $\mathcal{F}_\beta$. At the initial temperature, the entropy of $Q_\beta(\boldsymbol{S})$ is intended to be maximized rather than the Q function (the first term of equation (3.36)); therefore $Q_\beta(\boldsymbol{S})$ becomes closer to uniform distribution. While the temperature is decreasing, the form of $Q_\beta(\boldsymbol{S})$ changes from uniform to the original posterior and at the final temperature $1/\beta = 1$, the negative free energy $\mathcal{F}_\beta$ becomes equal to the lower bound $\mathcal{F}$, accordingly the DAEM algorithm agrees with the original EM algorithm.

If the distribution $Q_\beta(\boldsymbol{S})$ have complete flexibility, the optimal distribution which maximizes $\mathcal{F}_\beta$ is given by

$$Q_\beta(\boldsymbol{S}) = \frac{1}{Z_\beta}P(\boldsymbol{O},\boldsymbol{S}|\Lambda)^\beta = \frac{P(\boldsymbol{O},\boldsymbol{S}|\Lambda)^\beta}{\sum_{S} P(\boldsymbol{O},\boldsymbol{S}|\Lambda)^\beta} \qquad (3.38)$$

where Z_β is the normalization constant. In SL2D-HMMs, decoupled approximate distributions can be derived as

$$Q_\beta(\boldsymbol{S}^{(n)}) = \frac{1}{Z_\beta^{(n)}} P(\boldsymbol{S}^{(n)}|\Lambda)^\beta \prod_{t^{(n)}=1}^{T^{(n)}} h(t^{(n)}, S_{t^{(n)}}^{(n)})^\beta \tag{3.39}$$

and the normalization constant is given by

$$Z_\beta^{(n)} = \sum_{\boldsymbol{S}^{(n)}} P(\boldsymbol{S}^{(n)}|\Lambda)^\beta \prod_{t^{(n)}=1}^{T^{(n)}} h(t^{(n)}, S_{t^{(n)}}^{(n)})^\beta \tag{3.40}$$

The expectations with respect to this distribution can also be calculated by the forward-backward algorithm with using $P(\boldsymbol{S}^{(n)}|\Lambda)^\beta$ and $h(t^{(n)}, S_{t^{(n)}}^{(n)})^\beta$ as the transition probabilities and the observation probabilities, respectively.

3.4 Disadvantage

It must be noted that the modeling accuracy of SL2D-HMMs 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. Moreover, SL2D-HMMs cannot deal with affine deformation completely. In other words, SL2D-HMMs cannot represent rotational variations. This is because the model structure composed of two independent (horizontal and vertical) Markov chains.

3.5 Summary

In this chapter, separable lattice 2-D hidden Markov models (SL2D-HMMs) have been defined, in which multiple hidden state sequences interact to model the observations on a lattice. it is focused on the case of 2-D lattices, with a horizontal and vertical Markov chain, and their application to modeling images. SL2D-HMMs can perform an elastic matching in both horizontal and vertical directions; this makes it possible to model invariances to the size and location of an object. A training algorithm for SL2D-HMMs based on a variational approximation have been presented. Moreover, the deterministic annealing EM (DAEM) algorithm have been applied to the training of SL2D-HMMs with a variational approximation. However, the modeling accuracy of SL2D-HMMs is still insufficient. The next chapter will describe an extension of SL2D-HMMs for rotational variations. In Chapter 5 reformulates SL2D-HMMs by imposing explicit relationship between static and dynamic features to overcome these shortcomings.

Chapter 4

An extension of separable lattice 2-D HMMs for rotational variations

In the previous chapter, the model definition and the training algorithm for SL2D-HMMs were described. Although SL2D-HMMs can perform an elastic matching in both horizontal and vertical directions, SL2D-HMMs cannot deal with rotational variations. This chapter derives an extension of separable lattice 2-D HMMs for rotational variations. The training algorithm for the proposed model based variational EM algorithm is also derived.

4.1 Model structure representing rotational variations

To reduce the complexity, SL2D-HMMs have only one state sequence in each direction; this means that all horizontal/vertical lines of an observation lattice have the same state alignment for each direction. However, to represent the rotational variations, the models should have a different state alignment for each observation line and horizontal/vertical state alignments should be changed along with vertical/horizontal direction. In this thesis, we propose a new model structure with additional HMM states which represent the shifts of the state alignments of observation lines in a particular direction. Since the degree of the shift is controlled by the Markov chains, the proposed model can represent the dependency of the state alignments between consecutive observation lines. Therefore, the proposed model can perform a continuous elastic matching including rotational transformations. Figure 4.1 and 4.2 show the model structure of the proposed model and graphical representation for the proposed model, respectively.

The likelihood function of the proposed model is defined as follows:

$$\begin{aligned} P(\boldsymbol{O}, \boldsymbol{S}, \boldsymbol{d} \mid \Lambda) &= P(\boldsymbol{O} \mid \boldsymbol{S}, \boldsymbol{d}, \Lambda) \cdot P(\boldsymbol{S} \mid \Lambda) \cdot P(\boldsymbol{d} \mid \Lambda) \\ &= \prod_{\boldsymbol{t}} P(\boldsymbol{O_t} \mid \boldsymbol{S_{\bar{t}}}, \boldsymbol{d_t}, \Lambda) \prod_m P(\boldsymbol{S}^{(m)} \mid \Lambda) \cdot \prod_m P(\boldsymbol{d}^{(m)} \mid \Lambda) \end{aligned} \tag{4.1}$$

where $\boldsymbol{S}$ represents the reference state sequences corresponding to the state sequences of SL2D-HMMs and $\boldsymbol{d}$ represents the shift state sequences and consists of two Markov chains for each dimension:

$$\boldsymbol{d} = \left\{\boldsymbol{d}^{(1)}, \boldsymbol{d}^{(2)}\right\} \tag{4.2}$$

$$\boldsymbol{d}^{(m)} = \left\{d_1^{(m)},\ d_2^{(m)},\ \ldots,\ d_{T^{(n)}}^{(m)}\right\} \tag{4.3}$$

$$d_{t^{(n)}}^{(m)} \in \left\{D_{min}^{(m)},\ D_{min}^{(m)} + 1,\ \ldots,\ D_{max}^{(m)}\right\},\ n \neq m \tag{4.4}$$

where $D_{min}^{(m)}$ and $D_{max}^{(m)}$ represent the minimum and maximum shift of the m-th coordinate respectively, and $\boldsymbol{S_{\bar{t}}}$ is the shifted state defined as

$$\boldsymbol{S_{\bar{t}}} = \left(S_{\bar{t}^{(1)}}^{(1)},\ S_{\bar{t}^{(2)}}^{(2)}\right) = \left(S_{t^{(1)}+d_{t^{(2)}}^{(1)}}^{(1)},\ S_{t^{(2)}+d_{t^{(1)}}^{(2)}}^{(2)}\right) \tag{4.5}$$

where the following boundary conditions are assumed:

$$S_{\bar{t}^{(m)}}^{(m)} = \begin{cases} 1 & \left(\bar{t}^{(m)} \leq 0\right) \\ K^{(m)} & \left(\bar{t}^{(m)} > T^{(m)}\right) \end{cases} \tag{4.6}$$

Figure 4.3 shows an example of the state alignment of the proposed model where monotonic alignment can be obtained by using shift states.

Model parameters of the proposed model are summarized as follows:

- **Parameters for state transition probability of reference states $\boldsymbol{S}$**:

  1) $\Pi_S^{(m)} = \{\pi_{S,i}^{(m)} | 1 \leq i \leq K^{(m)}\}$: the initial state probability distribution, where $\pi_{S,i}^{(m)} = P(S_1^{(m)} = i|\Lambda)$ is the probability of state i at $t^{(m)} = 1$ in the m-th state sequence $\boldsymbol{S}^{(m)}$.

  2) $\boldsymbol{A}_S^{(m)} = \{a_{S,ij}^{(m)} \mid 1 \leq i, j \leq K^{(m)}\}$: the transition probability matrix, where $a_{S,ij}^{(m)} = P(S_{t^{(m)}}^{(m)} = j|S_{t^{(m)}-1}^{(m)} = i, \Lambda)$ is the transition probability from state i to state j in the m-th state sequence $\boldsymbol{S}^{(m)}$.

- **Parameters for state transition probability of shift states $\boldsymbol{d}$** :

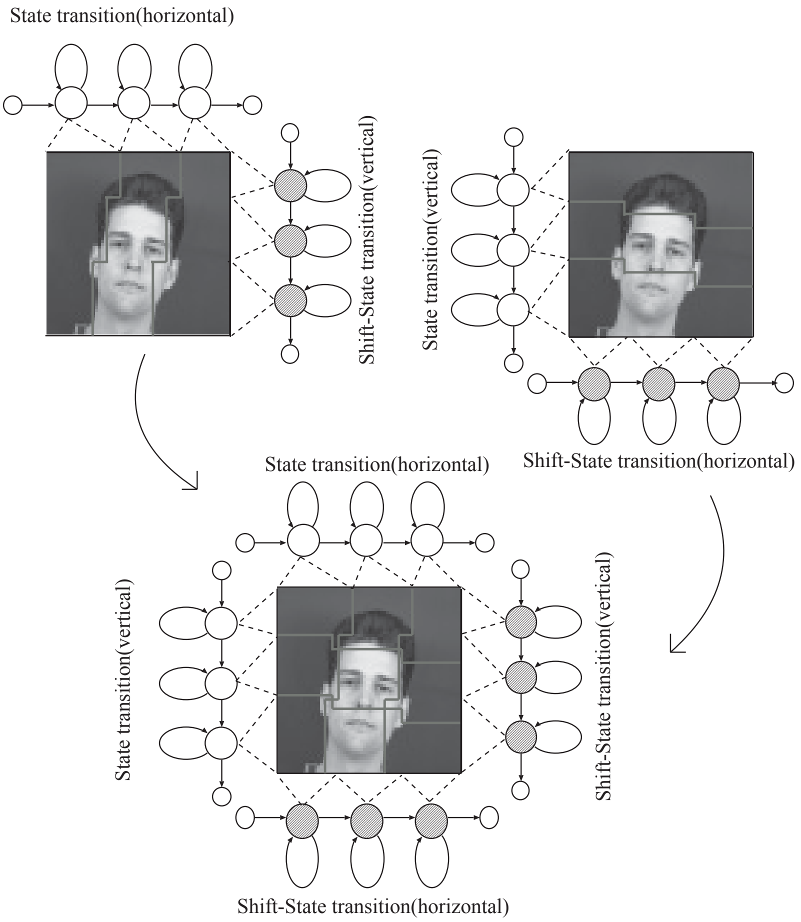

State transition(horizontal)
Shift-State transition(vertical)
State transition(vertical)
State transition(horizontal)
Shift-State transition(horizontal)
State transition(vertical)
Shift-State transition(vertical)
Shift-State transition(horizontal)

Figure 4.1: Model structure of the proposed model: The horizontal/vertical state alignments is changed along with vertical/horizontal state direction to represent the rotational variations.

1) $\Pi_d^{(m)} = \{\pi_{d,i}^{(m)} | 1 \le i \le K_d^{(m)}\}$: the initial state probability distribution, where $\pi_{d,i}^{(m)} = P(d_1^{(m)} = i|\Lambda)$ is the probability of state i at $t^{(n)} = 1$ in the m-th state sequence $\boldsymbol{d}^{(m)}$.

2) $\boldsymbol{A}_d^{(m)} = \{a_{d,ij}^{(m)} | D_{min}^{(m)} \le i, j \le D_{max}^{(m)}\}$: the transition probability matrix, where $a_{d,ij}^{(m)} = P(d_{t^{(n)}}^{(m)} = j | d_{t^{(n)}-1}^{(m)} = i, \Lambda)$ is the transition probability from state i to state j in the m-th state sequence $\boldsymbol{d}^{(m)}$.

- **Parameters for output probability distribution** :
  $\boldsymbol{B} = \{b_{\boldsymbol{k}}(\boldsymbol{O_t}) | \boldsymbol{k} \in \boldsymbol{K}\}$: the output probability distributions, where $b_{\boldsymbol{k}}(\boldsymbol{O_t})$ is the probability of observation vector $\boldsymbol{O_t}$ at the state $\boldsymbol{k}$ on the state lattice $\boldsymbol{K}$ and assumed to be a single Gaussian distribution : $P(\boldsymbol{O_t}|\boldsymbol{S_t} = \boldsymbol{k}) = \mathcal{N}(\boldsymbol{O_t}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$ where $\boldsymbol{\mu_k}$ and $\boldsymbol{\Sigma_k}$ are the mean vector and the covariance matrix, respectively.

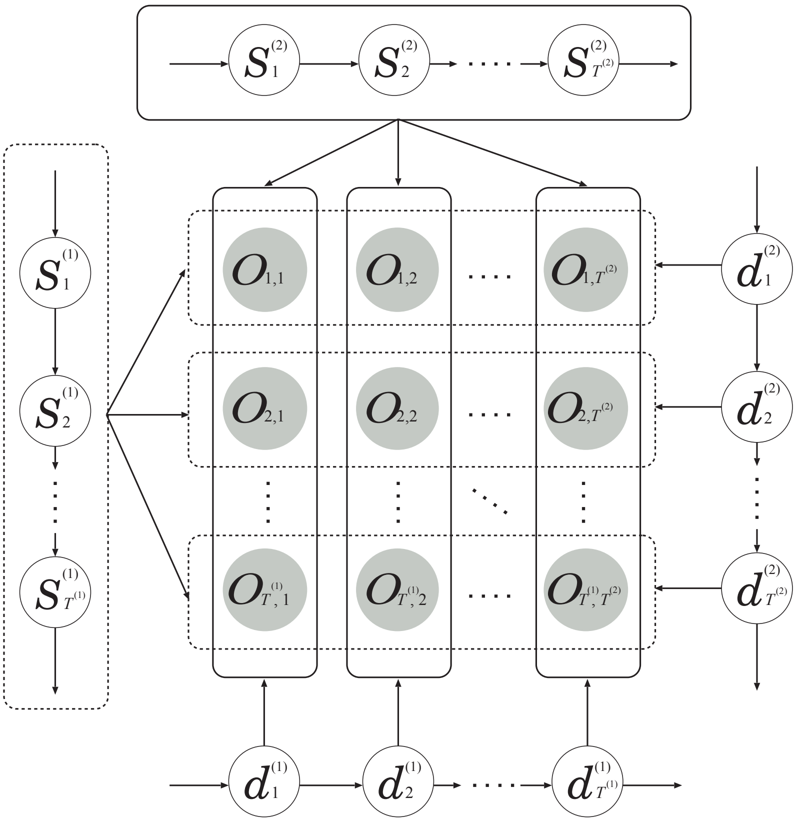


Figure 4.2: Graphical representation of the proposed model: The shift sequence affects the all data on the same observed line.

Using the above shorthand notation, the proposed model is defined as

$$\begin{aligned}\Lambda &= \{\Lambda_S^{(1)}, \Lambda_S^{(2)}, \Lambda_d^{(1)}, \Lambda_d^{(2)}, \boldsymbol{B}\}, && (4.7)\\ \Lambda_S^{(m)} &= \{\Pi_S^{(m)}, \boldsymbol{A}_S^{(m)}\}, && (4.8)\\ \Lambda_d^{(m)} &= \{\Pi_d^{(m)}, \boldsymbol{A}_d^{(m)}\}. && (4.9)\end{aligned}$$

The proposed model has potential to perform an continuous elastic matching beyond rotational variations. However, in this dissertation, the topology and the shift amounts are constrained to a special form which is expected to represent the continuous rotational variations. The example of the form for the m-th dimension where $D_{min}^{(m)} = -2$ and $D_{max}^{(m)} = 2$ is shown in figure 4.4.

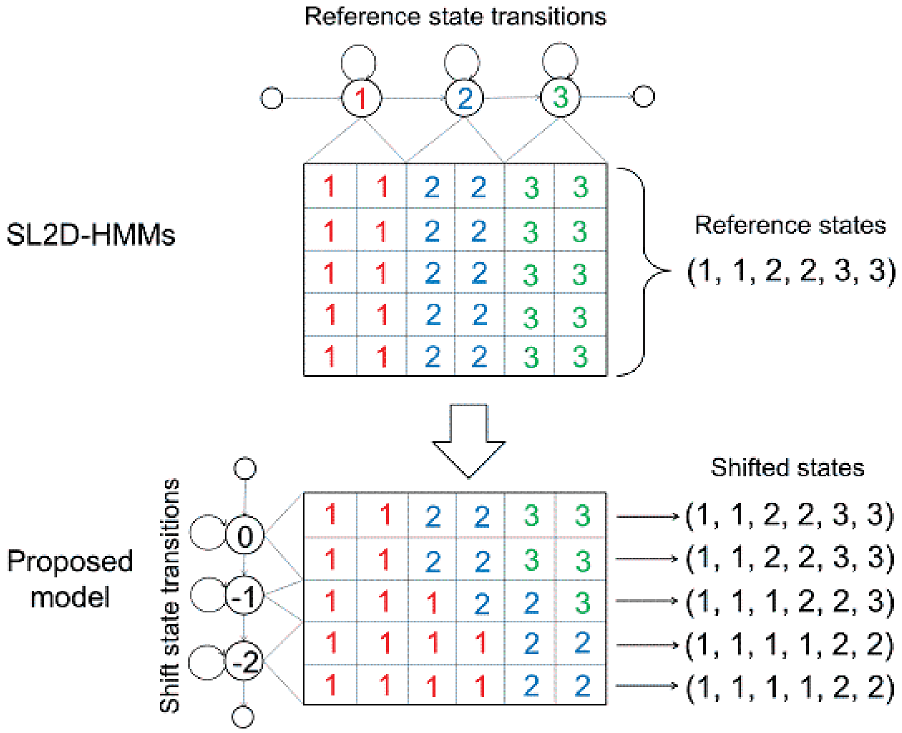


Figure 4.3: An example of state alignment of the proposed model for reference states and shifted states in horizontal direction: Without shift states (SL2D-HMMs), rectangle state alignments can be obtained while with shift states, monotonically shifted state alignments can be obtained in the proposed model.

4.2 Training algorithm

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

$$Q(\Lambda, \Lambda') = \sum_{\boldsymbol{S},\boldsymbol{d}} P(\boldsymbol{S}, \boldsymbol{d} \mid \boldsymbol{O}, \Lambda) \ln P(\boldsymbol{O}, \boldsymbol{S}, \boldsymbol{d} \mid \Lambda') \tag{4.10}$$

By maximizing the Q-function with respect to model parameters Λ, the re-estimation formula in the M-step can be easily derived. However, the calculation of the posterior distribution $P(\boldsymbol{S}, \boldsymbol{d} \mid \boldsymbol{O}, \Lambda)$ in the E-step is computationally intractable due to the combination of hidden variables. To derive a feasible problem, we applied the variational EM algorithm [18] to the training algorithm of the proposed model.

The variational methods approximate the posterior distribution over the hidden variables

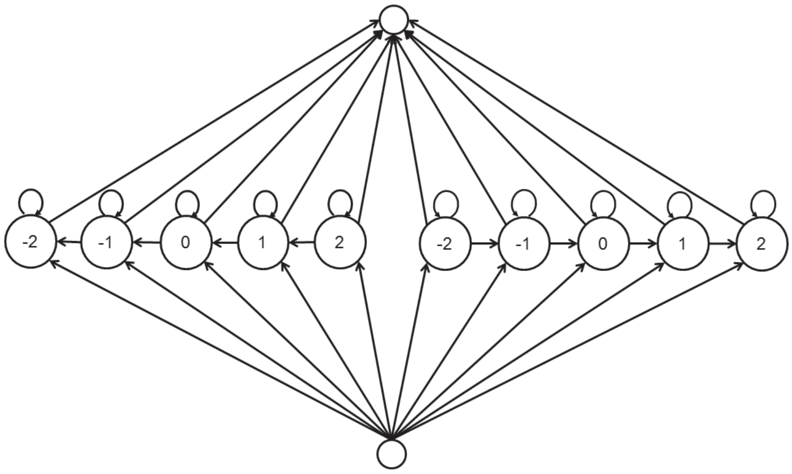


Figure 4.4: The example of topology of the transition probabilities of the m-th dimension shift states where $D^{(m)}_{min} = -2$ and $D^{(m)}_{max} = 2$; from this topology, monotonically increasing or decreasing sequence of the shift amount can be obtained and clockwise or counter-clockwise rotational variations can be represented.

by a tractable distribution. Any distribution over the hidden variables defines a lower bound on the log-likelihood

$$\begin{aligned}\ln P(\boldsymbol{O} \mid \Lambda) &= \ln \sum_{\boldsymbol{S}} \sum_{\boldsymbol{d}} Q(\boldsymbol{S}, \boldsymbol{d}) \frac{P(\boldsymbol{O}, \boldsymbol{S}, \boldsymbol{d} \mid \Lambda)}{Q(\boldsymbol{S}, \boldsymbol{d})} \\ &\geq \sum_{\boldsymbol{S}} \sum_{\boldsymbol{d}} Q(\boldsymbol{S}, \boldsymbol{d}) \ln \frac{P(\boldsymbol{O}, \boldsymbol{S}, \boldsymbol{d} \mid \Lambda)}{Q(\boldsymbol{S}, \boldsymbol{d})} \\ &= \mathcal{F}(Q, \Lambda) \end{aligned} \tag{4.11}$$

where Jensen's inequality has been applied. The difference between $\ln P(\boldsymbol{O} \mid \Lambda)$ and $\mathcal{F}$ is given by the KL divergence between $Q(\boldsymbol{S}, \boldsymbol{d})$ and the posterior distribution of the hidden variables $P(\boldsymbol{S}, \boldsymbol{d} \mid \boldsymbol{O}, \Lambda)$:

$$\begin{aligned}\mathcal{F}(Q, \Lambda) &= \sum_{\boldsymbol{S}} \sum_{\boldsymbol{d}} Q(\boldsymbol{S}, \boldsymbol{d}) \ln \frac{P(\boldsymbol{O}, \boldsymbol{S}, \boldsymbol{d} \mid \Lambda)}{Q(\boldsymbol{S}, \boldsymbol{d})} \\ &= \sum_{\boldsymbol{S}} \sum_{\boldsymbol{d}} Q(\boldsymbol{S}, \boldsymbol{d} | \Lambda) \ln P(\boldsymbol{O} \mid \Lambda) + \sum_{\boldsymbol{S}} \sum_{\boldsymbol{d}} Q(\boldsymbol{S}, \boldsymbol{d}) \ln \frac{P(\boldsymbol{S}, \boldsymbol{d} \mid \boldsymbol{O}, \Lambda)}{Q(\boldsymbol{S}, \boldsymbol{d})} \\ &= \ln P(\boldsymbol{O} \mid \Lambda) - \mathrm{KL}(Q \,||\, P) \end{aligned} \tag{4.12}$$

Since the true log-likelihood $\ln P(\boldsymbol{O} \mid \Lambda)$ is independent of $Q(\boldsymbol{S}, \boldsymbol{d})$, maximizing the lower bound $\mathcal{F}$ is equivalent to minimizing the KL divergence. If we allow $Q(\boldsymbol{S}, \boldsymbol{d})$ to have complete flexibility then we see that the optimal $Q(\boldsymbol{S}, \boldsymbol{d})$ distribution is given by the true posterior $P(\boldsymbol{S}, \boldsymbol{d} | \boldsymbol{O}, \Lambda)$, in the case where the KL divergence is zero and the bound becomes exact. In order to yield a tractable algorithm, it is necessary to consider a more restricted structure of $Q(\boldsymbol{S}, \boldsymbol{d})$ distributions. Given the structure, the parameters of $Q(\boldsymbol{S}, \boldsymbol{d})$ are varied so as to obtain the tightest possible bound, which maximizes $\mathcal{F}$.

The variational EM algorithm iteratively maximizes $\mathcal{F}$ with respect to the Q and Λ holding the other parameters fixed:

$$\begin{array}{rcrcl} \text{(E-step)} & : & Q^{(k+1)} & = & \arg\max_{Q \in C} \mathcal{F}(Q, \Lambda^{(k)}) \\ \text{(M-step)} & : & \Lambda^{(k+1)} & = & \arg\max_{\Lambda} \mathcal{F}(Q^{(k+1)}, \Lambda) \end{array}$$

where C is the set of constrained distributions. In this procedure, the lower bound $\mathcal{F}$ is guaranteed to increase instead of the value of the Q-function.

The complexity and the approximation property of the variational EM algorithm are dependent on a constraint to the posterior distribution $Q(\boldsymbol{S}, \boldsymbol{d})$ and it should be determined for each structure of graphical models. Here we consider a constrained family of variational distributions for the proposed model by assuming that $Q(\boldsymbol{S}, \boldsymbol{d})$ factorizes over subset $\boldsymbol{S}^{(m)}$ and $\boldsymbol{d}^{(m)}$ of the variables in $\boldsymbol{S}$ and $\boldsymbol{d}$, so that

$$\begin{aligned} Q(\boldsymbol{S}, \boldsymbol{d}) &= Q(\boldsymbol{S})Q(\boldsymbol{d}) &(4.13)\\ &= \prod_{m=1}^{M} Q(\boldsymbol{S}^{(m)}) \prod_{m=1}^{M} Q(\boldsymbol{d}^{(m)}) &(4.14) \end{aligned}$$

where $Q(\boldsymbol{S})$ and $Q(\boldsymbol{d})$ are the posterior distribution over $\boldsymbol{S}$ and $\boldsymbol{d}$, respectively. Also, $\sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) = 1$ and $\sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{d}^{(m)}) = 1$, $m = 1, \ldots, M$. The optimal distributions of the subsets are obtained by maximizing $\mathcal{F}$ independently while keeping the other distributions fixed:

$$Q(\boldsymbol{S}^{(m)}) \propto P(\boldsymbol{S}^{(m)} \mid \Lambda) \exp\left[\sum_{\boldsymbol{d}} Q(\boldsymbol{d}) \sum_{\boldsymbol{S} \backslash \boldsymbol{S}^{(m)}} \prod_{n \neq m} Q(\boldsymbol{S}^{(n)}) \ln P(\boldsymbol{O} \mid \boldsymbol{S}, \boldsymbol{d}, \Lambda)\right] \tag{4.15}$$

$$Q(\boldsymbol{d}^{(m)}) \propto P(\boldsymbol{d}^{(m)} \mid \Lambda) \exp\left[\sum_{\boldsymbol{S}} Q(\boldsymbol{S}) \sum_{\boldsymbol{d} \backslash \boldsymbol{d}^{(m)}} \prod_{n \neq m} Q(\boldsymbol{d}^{(n)}) \ln P(\boldsymbol{O} \mid \boldsymbol{S}, \boldsymbol{d}, \Lambda)\right] \tag{4.16}$$

The detail of the derivation will be described in appendix A.1. The E-step consists of the updates of $Q(\boldsymbol{S}^{(1)})$, $Q(\boldsymbol{S}^{(2)})$, $Q(\boldsymbol{d}^{(1)})$ and $Q(\boldsymbol{d}^{(2)})$, which interact through the expectations. By inspection, the distribution $Q(\boldsymbol{S}^{(1)})$, $Q(\boldsymbol{S}^{(2)})$, $Q(\boldsymbol{d}^{(1)})$ and $Q(\boldsymbol{d}^{(2)})$ have the same structure as the posterior of standard HMMs. Therefore, the forward-backward algorithm can be used to compute the following expectations efficiently:

$$
\begin{aligned}
\left\langle \left(S^{(m)}_{t^{(m)}}, i \right) \right\rangle &= \sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) \delta(S^{(m)}_{t^{(m)}}, i) && (4.17) \\
\left\langle \left(S^{(m)}_{t^{(m)}-1}, i \right) \left(S^{(m)}_{t^{(m)}}, j \right) \right\rangle &= \sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) \delta(S^{(m)}_{t^{(m)}-1}, i) \delta(S^{(m)}_{t^{(m)}}, j) && (4.18) \\
\left\langle \left(d^{(m)}_{t^{(n)}}, i \right) \right\rangle &= \sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{d}^{(m)}) \delta(d^{(m)}_{t^{(n)}}, i) && (4.19) \\
\left\langle \left(d^{(m)}_{t^{(n)}-1}, i \right) \left(d^{(m)}_{t^{(n)}}, j \right) \right\rangle &= \sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{d}^{(m)}) \delta(d^{(m)}_{t^{(n)}-1}, i) \delta(d^{(m)}_{t^{(n)}}, j) && (4.20) \\
\left\langle \left(S^{(m)}_{t^{(m)}+d^{(m)}_{t^{(n)}}}, k^{(m)} \right) \left(d^{(m)}_{t^{(n)}}, l^{(m)} \right) \right\rangle &= \sum_{\boldsymbol{S}^{(m)}} \sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{S}^{(m)}) Q(\boldsymbol{d}^{(m)}) \times \\
&\quad \delta(S^{(m)}_{t^{(m)}+d^{(m)}_{t^{(n)}}}, k^{(m)}) \delta(d^{(m)}_{t^{(n)}}, l^{(m)}) && (4.21) \\
\langle (\boldsymbol{S_t}, \boldsymbol{k})(\boldsymbol{d_t}, \boldsymbol{l}) \rangle &= \prod_m \left\langle \left(S^{(m)}_{t^{(m)}+d^{(m)}_{t^{(n)}}}, k^{(m)} \right) \left(d^{(m)}_{t^{(n)}}, l^{(m)} \right) \right\rangle \\
& && (4.22)
\end{aligned}
$$

where $n \neq m$. Using these expectations, the re-estimation formula of the proposed model in the M-step are derived as follows.

$$
\begin{aligned}
\pi^{(m)}_{S,i} &= \left\langle \left(S^{(m)}_{1}, i \right) \right\rangle && (4.23) \\
\pi^{(m)}_{d,i} &= \left\langle \left(d^{(m)}_{1}, i \right) \right\rangle && (4.24) \\
a^{(m)}_{S,ij} &= \frac{\sum_{t^{(m)}=2}^{T^{(m)}} \left\langle \left(S^{(m)}_{t^{(m)}-1}, i \right) (S^{(m)}_{t^{(m)}}, j) \right\rangle}{\sum_{t^{(m)}=1}^{T^{(m)}} \left\langle \left(S^{(m)}_{t^{(m)}}, i \right) \right\rangle} && (4.25) \\
a^{(m)}_{d,ij} &= \frac{\sum_{t^{(n)}=2}^{T^{(n)}} \left\langle \left(d^{(m)}_{t^{(n)}-1}, i \right) (d^{(m)}_{t^{(n)}}, j) \right\rangle}{\sum_{t^{(n)}=1}^{T^{(n)}} \left\langle \left(d^{(m)}_{t^{(n)}}, i \right) \right\rangle} && (4.26) \\
\boldsymbol{\mu_k} &= \frac{\sum_{\boldsymbol{t}} \sum_{\boldsymbol{l}} \langle (\boldsymbol{S_t}, \boldsymbol{k})(\boldsymbol{d_t}, \boldsymbol{l}) \rangle \boldsymbol{O_t}}{\sum_{\boldsymbol{t}} \sum_{\boldsymbol{l}} \langle (\boldsymbol{S_t}, \boldsymbol{k})(\boldsymbol{d_t}, \boldsymbol{l}) \rangle} && (4.27)
\end{aligned}
$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{t,l} \langle (\boldsymbol{S}_t, \boldsymbol{k})(\boldsymbol{d}_t, \boldsymbol{l}) \rangle (\boldsymbol{O}_t - \boldsymbol{\mu}_k)(\boldsymbol{O}_t - \boldsymbol{\mu}_k)^\top}{\sum_{t,l} \langle (\boldsymbol{S}_t, \boldsymbol{k})(\boldsymbol{d}_t, \boldsymbol{l}) \rangle} \tag{4.28}$$

The derivation of the above formulas will be described in appendix A.2.

4.3 Experiments

4.3.1 Experimental conditions

To demonstrate the modeling ability of the proposed model, face recognition experiments on the XM2VTS database [35] were conducted. we prepared eight images of 100 subjects; seven images are used for training and one image for testing. The face images were extracted form the original images (720×576 pixels and transformed into gray-scale) and then sub-sampled to 64 × 64 pixels. In this process, we prepared four sets of data:

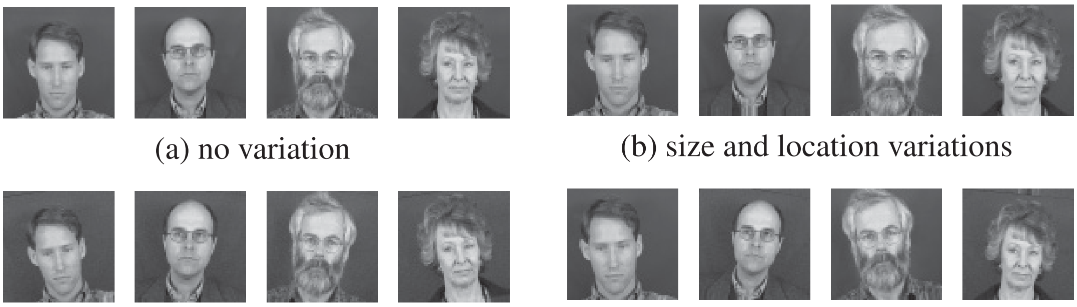
- "dataset 1" : the size- and location-normalized data. The original database does not include much variations of size and location, hence the center of the original images was used as the face location and the size was fixed to 550 × 550 pixels.

- "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 × 20 pixels from the center and the range of size 500 × 500 ∼ 600 × 600 with fixed aspect.

- "dataset 3" : data with rotational variations. The rotation angles are randomly generated within −10 ∼ 10 degrees from Gaussian distribution with 0.0 mean and 5.0 standard deviation.

- "dataset 4" : data with size, location and rotational variations. The size and location variations were generated as well as "dataset 2" and the rotational variations were generated as well as "dataset 3".

Figure 4.5 shows the examples of four datasets. Although it was already confirmed that the recognition performance was significantly improved with appropriate feature vectors



(a) no variation

(b) size and location variations

(c) rotational variations

(d) size, location and rotational variations

Figure 4.5: Examples of training data; with no variation (a) and with variations of size and location (b), with rotational variations (c) and with variations of size, location and rotations (d).

such as 2-D discrete cosine transform coefficients or linear regression coefficients of images, the pixel intensity values were used as features in this dissertation. 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 rotational variations. For the purpose of improving the recognition performance, the SL2D-HMMs were extended by integrating with a linear feature extraction such as probabilistic PCA or factor analyzers [10]. In the dissertation, 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 dissertation can be easily integrated with a linear feature extraction as [10] for improving recognition performance.

The number of reference states was 24×24 and the number of shift states was varied among 6×6, 10×10, 14×14, 18×18 and 22×22, corresponding to the conditions that $-D_{min}^{(m)} = D_{max}^{(m)} = 1, 2, 3, 4$ and 5, respectively. The number of reference states was previously optimized to give the best recognition performance on SL2D-HMMs. The transition probabilities for each sequence of reference states were assumed to be a left-to-right and top-to-bottom no skip topology and the transition probabilities for each sequence of shift states were assumed to be the topology as shown in figure 4.4.

4.3.2 Experimental results

Recognition performance

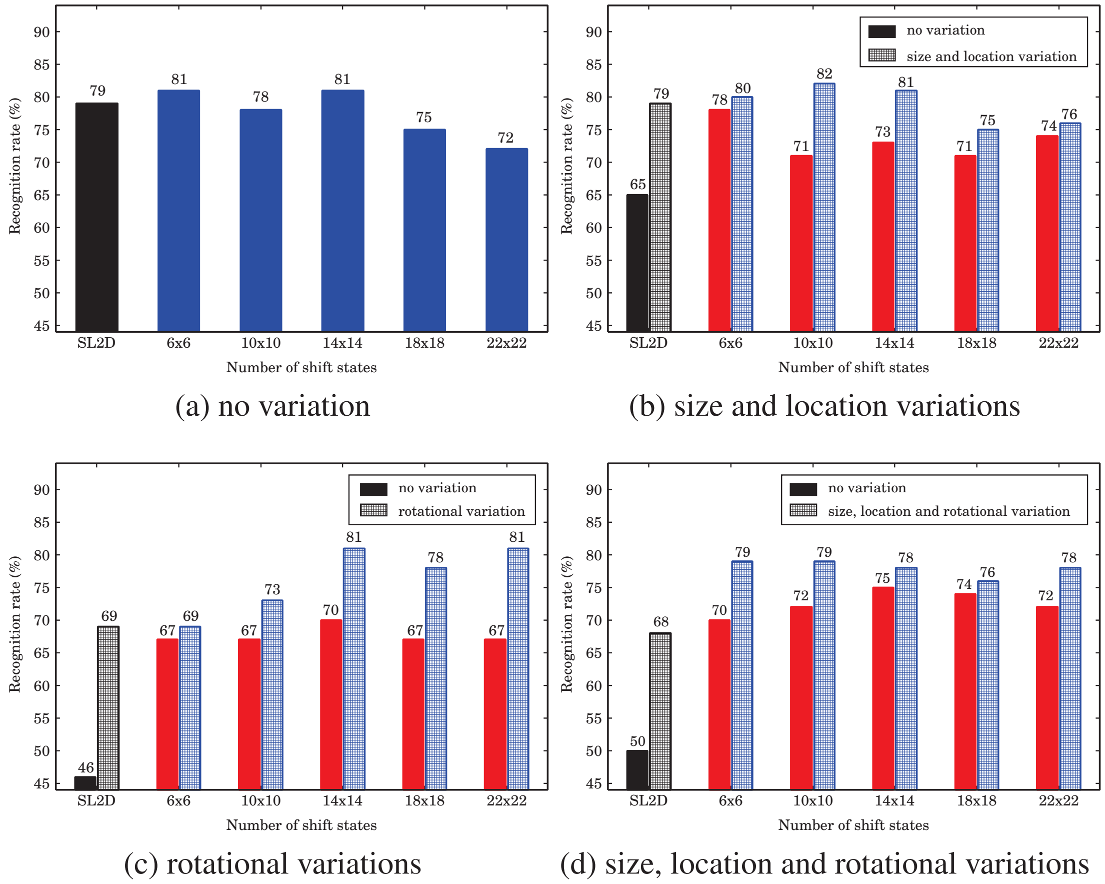
Figure 4.6(a), 4.6(b), 4.6(c) and 4.6(d) show the recognition rates of the test dataset with no variation (a), with variations of size and location (b), with rotational variations (c) and with variations of size, location and rotations (d), respectively. In the figures, plain boxes and meshed ones represent the recognition rates of the models trained from the dataset with no variation and the same variation as the test dataset, respectively.

From figure 4.6(b), it can be seen that the proposed model possesses the comparable normalization ability to the SL2D-HMMs for size and location variations. Also, from figure 4.6(c), it can be seen that SL2D-HMMs degrade the recognition performance when they were trained and tested on “dataset3” where rotational variations were included, while the proposed model improves the recognition performance significantly compared with the SL2D-HMMs (meshed boxes). Especially, the highest recognition rate of 81% was obtained at 14 × 14 and 22 × 22 shift states, which is comparable to the recognition rate of SL2D-HMMs on “dataset 1.” This means that the proposed model can normalize rotational variations appropriately. It also can be seen that the proposed model improves the performance to rotational variations from figure 4.6(d) (meshed boxes). Particularly, the recognition rates of 79% at 6 × 6, 10 × 10, 14 × 14 and 22 × 22 shift states were obtained, which also indicates that the proposed model can normalize not only the size and location variations but also the rotational variations accurately.

Comparing the models trained from no variation datasets (plain boxes) and matched variation datasets (meshed boxes), the recognition rates of the matched variation were higher than those of the no variation datasets, even though no variation datasets were appropriately normalized. This is because the models over-fitted to the variation of the training datasets. However, from another point of the view, the proposed model can preserve the information of variation in the training data. It might be useful for some classification tasks, e.g., the model can use a kind of information that some target objects tend to rotate and the others are not for classification.

State alignments

Figure 4.7 and figure 4.8 show the examples of mean vectors of SL2D-HMMs and the proposed model, and the visualized state alignments obtained by the Viterbi algorithm, respectively. In figure 4.7, the number of shift states of the proposed model is 22 × 22. The mean vectors were estimated from “dataset 1,” “dataset 2,” “dataset 3,” and “dataset



(a) no variation

(b) size and location variations

(c) rotational variations

(d) size, location and rotational variations

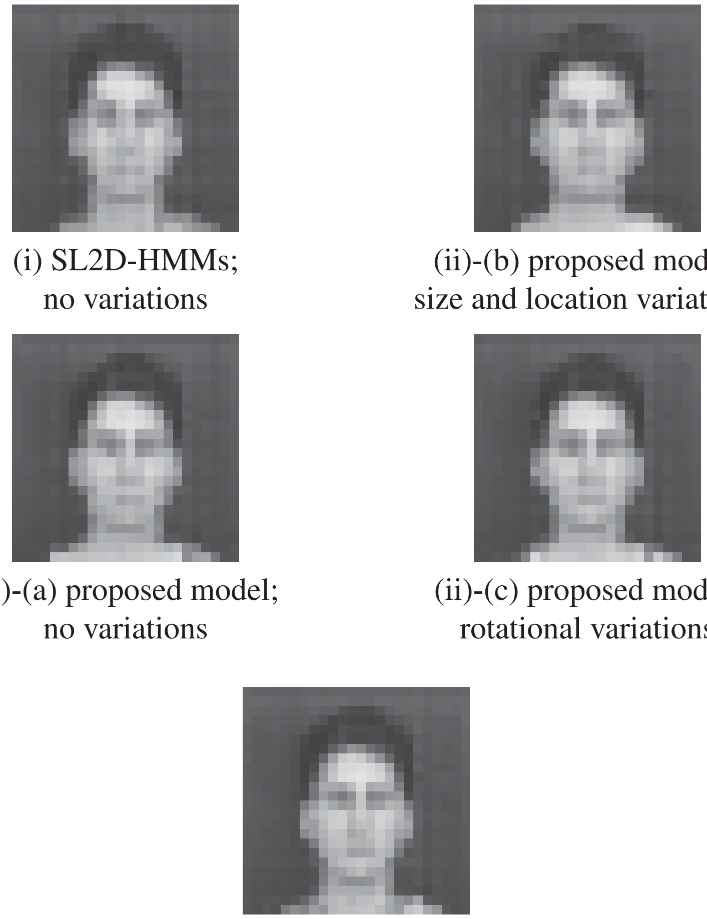
Figure 4.6: Recognition rates of the SL2D-HMMs and proposed model for each shift states tested on the dataset with no variation (a), with variations of size and location (b), with rotational variations (c) and with variations of size, location and rotations (d), respectively. In the figures, plain boxes and meshed ones represent the recognition rates of the models trained from the dataset with no variation and the same variation as the test dataset, respectively.

4," respectively. The state alignments are represented by the mean vectors of the states corresponding to the observations of the test data. The values below the images represent the averaged log-likelihoods of the observation per pixel given the best 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 that more preferable matching was obtained in terms of the maximum likelihood criterion.

From the results, we can observe that SL2D-HMMs could not deal with the rotational variations due to the constraint of the model structure. The likelihood of the test data was also significantly decreased with increasing the rotational angle of the test data. Contrary to this, when the rotational angle of the test data was -10, 0 or 10 degrees, the rotational variations of the data can be represented by the proposed model and the differences of the likelihood between 0 degree and 10, -10 degrees were smaller than those of the SL2D-HMMs. It seemed that the maximum value of the shift amount obtained by the proposed model was sufficient to represent the rotational angle ± 10 degrees. For the model (c) and (d), the maximum/minimum value of the rotational angle in the corresponding training dataset was between ± 10 degrees. This also led to the preferable results. On the other hand, when the rotational angle was larger, i.e. ± 20 degrees, the shift amount provided by the proposed model was not sufficient, so that the proposed model could not deal rotational variations compared to the results as the angle was ± 10. Similarly, the proper state alignment of the reference state was not obtained. This is because, as shown in eq. (4.15) and (4.16), the reference state sequences and the shift state sequences are dependent on each other through the variational distributions. Therefore it was difficult to estimate the proper reference state sequences once the improper shift state sequences were estimated from the test data. From these results, it was suggested that the number of shift states need to be determined according to the degree of rotational variation.

4.4 Summary

This chapter has derived an extension of separable lattice 2-D HMMs to deal with rotational data variations. The proposed model has additional HMM states which represent the shifts of the state alignments of observation lines in a particular direction. In face recognition experiments on the XM2VTS database, the proposed model achieved better results to the images than the conventional SL2D-HMMs. Moreover, the state alignments shows that the proposed model can normalize not only size and location variations but also rotational variations. The next chapter will derive a novel statistical model, named as separable lattice trajectory 2-D HMM by imposing explicit relationship between static



(i) SL2D-HMMs;
no variations

(ii)-(b) proposed model;
size and location variations

(ii)-(a) proposed model;
no variations

(ii)-(c) proposed model;
rotational variations

(ii)-(d) proposed model;
size, location and rotational
variations

Figure 4.7: Example of mean vectors: (i) is the mean vectors of the SL2D-HMMs. (ii) is the mean vectors of the proposed model. The number of shift state of (ii) is 22×22. They were estimated from the normalized data ("dataset 1").

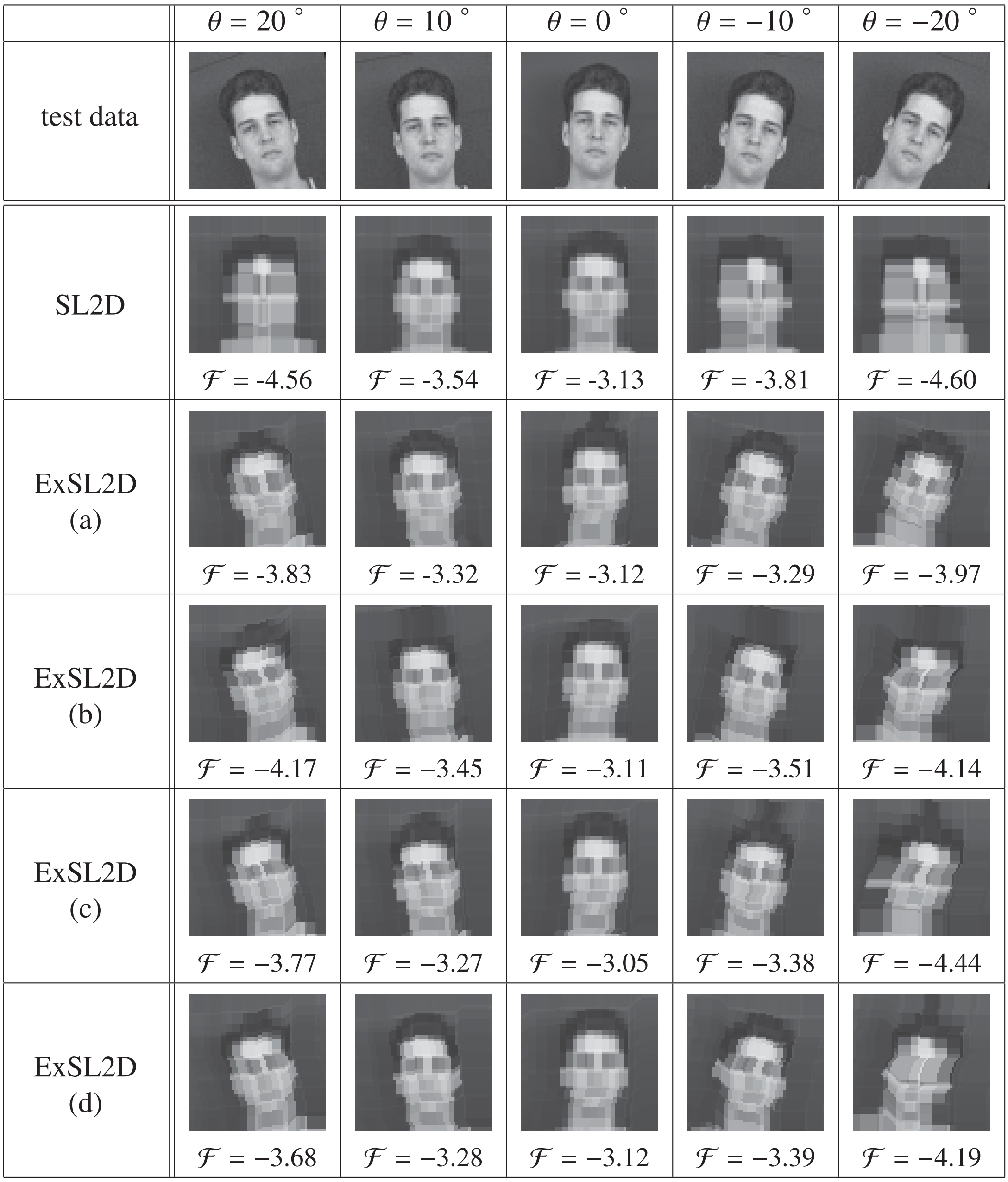


Figure 4.8: Examples of test data and the visualized state alignments on the dataset with no variation (a), with variations of size and location (b), with rotational variations (c) and with variations of size, location and rotations (d), respectively.The θ means the rotational angle for each test data. The $\mathcal{F}$ means the estimated log-likelihood to test data.

and dynamic features into separable lattice 2-D HMMs.

Chapter 5

Separable lattice trajectory 2-D HMMs

In Chapter 3, 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 [19]:

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

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) [20] to capture dependencies between adjacent observations can enhance the performance of the HMM-based speech recognizers [36]. 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 Section 5.2, the proposed model will be derived in order to avoid the above problem.

5.1 Applications of dynamic features

5.1.1 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 $\boldsymbol{o} = [\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T]$ be the sequence of speech parameter vectors, where $\boldsymbol{o}_t$ is a speech parameter vector at time t. In a typical speech recognition system, it is assumed that the speech parameter vector $\boldsymbol{o}_t$ is a $3M \times 1$ vector consisting of an M-dimensional acoustic static feature

$$\boldsymbol{c}_t = [c_t(1), c_t(2), \ldots, c_t(M)] \tag{5.1}$$

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

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]. \tag{5.2}$$

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

$$\Delta\boldsymbol{c}_t = \sum_{\tau=-L_-^{(1)}}^{L_+^{(1)}} w^{(1)}(\tau)\boldsymbol{c}_{t+\tau}, \tag{5.3}$$

$$\Delta^2\boldsymbol{c}_t = \sum_{\tau=-L_-^{(2)}}^{L_+^{(2)}} w^{(2)}(\tau)\boldsymbol{c}_{t+\tau}, \tag{5.4}$$

where $\{w^{(d)}(\tau)\}_{\tau=-L_-^{(d)},\ldots,L_+^{(d)}}$ 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 $\boldsymbol{o} = \left[\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top\right]^\top$ and static feature sequence $\boldsymbol{c} = \left[\boldsymbol{c}_1^\top, \boldsymbol{c}_2^\top \ldots, \boldsymbol{c}_T^\top\right]^\top$ can be arranged in a matrix form as

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}, \tag{5.5}$$

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

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_1 & \ldots & \boldsymbol{W}_t & \ldots & \boldsymbol{W}_T \end{bmatrix}^\top \otimes \boldsymbol{I}_{M\times M}, \tag{5.6}$$

$$\boldsymbol{W}_t = \left[\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}\right], \tag{5.7}$$

$$\begin{aligned}\boldsymbol{w}_t^{(d)} = \Big[&\underbrace{0, \ldots, 0}_{t-L_-^{(d)}-1}, w^{(d)}(-L_-^{(d)}), \ldots, w^{(d)}(0), \\ &\ldots, w^{(d)}(L_+^{(d)}), \underbrace{0, \ldots, 0}_{T-\left(t+L_+^{(d)}\right)}\Big]^\top, \quad d = 0, 1, 2\end{aligned} \tag{5.8}$$

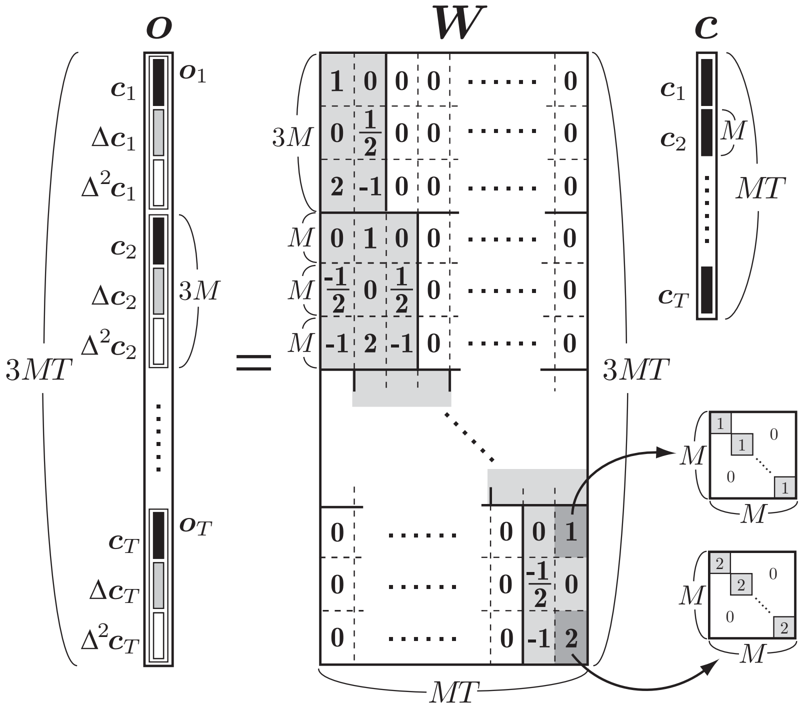


Figure 5.1: An example of relationship between the observation vector sequence $\boldsymbol{o}$ and the static feature vector sequence $\boldsymbol{c}$ in a matrix form [1], where the dynamic feature vectors are calculated using Eqs. (5.3) and (5.4) with $L_-^{(1)} = L_+^{(1)} = L_-^{(2)} = L_+^{(2)} = 1$, $w^{(1)}(-1) = -0.5$, $w^{(1)}(0) = 0.0$, $w^{(1)}(1) = 0.5$, $w^{(2)}(-1) = 1.0$, $w^{(1)}(0) = -2.0$, $w^{(2)}(1) = 1.0$.

where $L_-^{(0)} = L_+^{(0)} = 0$, $\boldsymbol{w}^{(0)} = 1$, and $\otimes$ denotes the Kronecker product for matrices. An example of the relationship is shown in Figure 5.1.

5.1.2 Dynamic features for image data

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

$$\boldsymbol{C}_t = [C_t(1), C_t(2), \ldots, C_t(M)]^\top \tag{5.9}$$

and horizontal/vertical dynamic feature vectors, $\Delta^{(H)}\boldsymbol{C}_t$ and $\Delta^{(V)}\boldsymbol{C}_t$, that is [1]

$$\boldsymbol{O}_t = \left[\boldsymbol{C}_t^\top, \Delta^{(H)}\boldsymbol{C}_t^\top, \Delta^{(V)}\boldsymbol{C}_t^\top\right]^\top, \tag{5.10}$$

[1] Using higher-order dynamic features is straightforward. Moreover, dynamic features in other directions, e.g., diagonal dynamic features can be adopted easily.

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

$$\Delta^{(H)}\boldsymbol{C}_{\boldsymbol{t}} = \sum_{\tau=-L_-^{(H)}}^{L_+^{(H)}} w^{(H)}(\tau)\boldsymbol{C}_{(t^{(1)}+\tau,t^{(2)})}, \tag{5.11}$$

$$\Delta^{(V)}\boldsymbol{C}_{\boldsymbol{t}} = \sum_{\tau=-L_-^{(V)}}^{L_+^{(V)}} w^{(V)}(\tau)\boldsymbol{C}_{(t^{(1)},t^{(2)}+\tau)}, \tag{5.12}$$

where $\left\{w^{(H)}(\tau)\right\}_{\tau=-L_-^{(H)},\ldots,L_+^{(H)}}$ and $\left\{w^{(V)}(\tau)\right\}_{\tau=-L_-^{(V)},\ldots,L_+^{(V)}}$ 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

$$\boldsymbol{O} = \left[\begin{array}{ccccc} \boldsymbol{O}_{(1,1)}^{\top} & \ldots & \boldsymbol{O}_{\boldsymbol{t}}^{\top} & \ldots & \boldsymbol{O}_{(T^{(1)},T^{(2)})}^{\top} \end{array}\right]^{\top}, \tag{5.13}$$

$$\boldsymbol{C} = \left[\begin{array}{ccccc} \boldsymbol{C}_{(1,1)}^{\top} & \ldots & \boldsymbol{C}_{\boldsymbol{t}}^{\top} & \ldots & \boldsymbol{C}_{(T^{(1)},T^{(2)})}^{\top} \end{array}\right]^{\top}, \tag{5.14}$$

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

A linear relationship between $\boldsymbol{O}$ and $\boldsymbol{C}$ in 2-D case, which is similar to Eq. (5.5) in 1-D case, can be obtained as

$$\boldsymbol{O} = \boldsymbol{W}\boldsymbol{C}, \tag{5.15}$$

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

$$\boldsymbol{W} = \left[\begin{array}{ccccc} \boldsymbol{W}_{(1,1)} & \ldots & \boldsymbol{W}_{\boldsymbol{t}} & \ldots & \boldsymbol{W}_{(T^{(1)},T^{(2)})} \end{array}\right]^{\top} \otimes \boldsymbol{I}_{M\times M}, \tag{5.16}$$

$$\boldsymbol{W}_{\boldsymbol{t}} = \left[\boldsymbol{w}_{\boldsymbol{t}}^{(S)}, \boldsymbol{w}_{\boldsymbol{t}}^{(H)}, \boldsymbol{w}_{\boldsymbol{t}}^{(V)}\right], \tag{5.17}$$

where $\boldsymbol{w}_{\boldsymbol{t}}^{(S)}$, $\boldsymbol{w}_{\boldsymbol{t}}^{(H)}$, and $\boldsymbol{w}_{\boldsymbol{t}}^{(V)}$ are $T^{(1)}T^{(2)}$ size vectors. They are defined so that following relationships are satisfied based on Eqs. (5.10), (5.11), (5.12) and (5.17):

$$\boldsymbol{C}_{\boldsymbol{t}} = \left(\boldsymbol{w}_{\boldsymbol{t}}^{(S)\top} \otimes \boldsymbol{I}_{M\times M}\right)\boldsymbol{C}, \tag{5.18}$$

$$\Delta\boldsymbol{C}_{\boldsymbol{t}}^{(H)} = \left(\boldsymbol{w}_{\boldsymbol{t}}^{(H)\top} \otimes \boldsymbol{I}_{M\times M}\right)\boldsymbol{C}, \tag{5.19}$$

$$\Delta\boldsymbol{C}_{\boldsymbol{t}}^{(V)} = \left(\boldsymbol{w}_{\boldsymbol{t}}^{(V)\top} \otimes \boldsymbol{I}_{M\times M}\right)\boldsymbol{C}, \tag{5.20}$$

$$\boldsymbol{O}_{\boldsymbol{t}} = \left(\boldsymbol{W}_{\boldsymbol{t}}^{\top} \otimes \boldsymbol{I}_{M\times M}\right)\boldsymbol{C}. \tag{5.21}$$

The functions of window vectors $\boldsymbol{w}_{\boldsymbol{t}}^{(S)}$, $\boldsymbol{w}_{\boldsymbol{t}}^{(H)}$, and $\boldsymbol{w}_{\boldsymbol{t}}^{(V)}$ can be explained as follows: From Eq. (5.18), $\boldsymbol{w}_{\boldsymbol{t}}^{(S)}$ is a vector which extract the static feature vector at $\boldsymbol{t} = \left(t^{(1)}, t^{(2)}\right)$ from image data. Furthermore, from Eqs. (5.19) and (5.20), $\boldsymbol{w}_{\boldsymbol{t}}^{(H)}$ and $\boldsymbol{w}_{\boldsymbol{t}}^{(V)}$ are vectors which extract the gradients of horizontal and vertical direction centered at $\boldsymbol{t}$, respectively. Examples of $\boldsymbol{w}_{\boldsymbol{t}}^{(S)}$, $\boldsymbol{w}_{\boldsymbol{t}}^{(H)}$, and $\boldsymbol{w}_{\boldsymbol{t}}^{(V)}$ are shown in Figure 5.2, where the maximum window length $L = 1$ and $M = 1$ for simplicity.

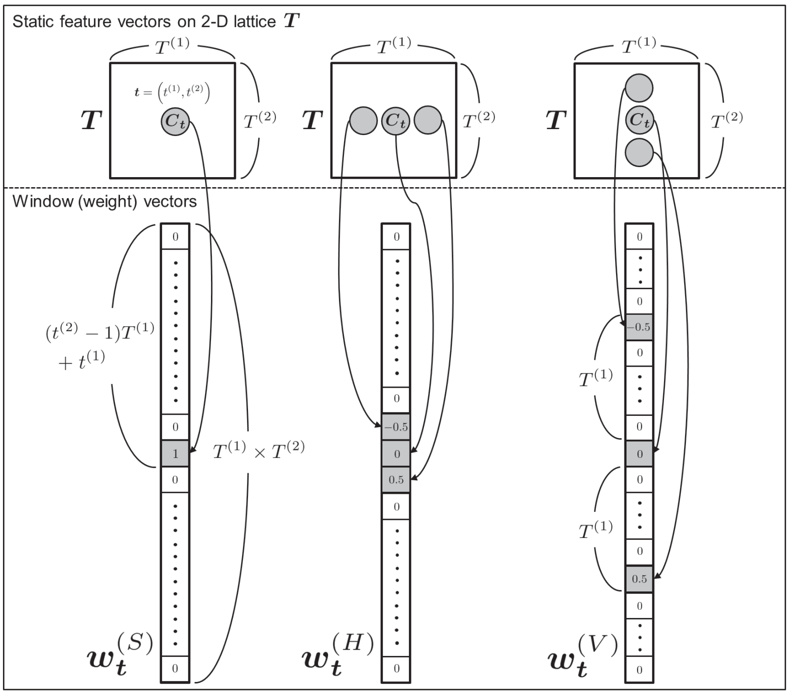

Static feature vectors on 2-D lattice $\boldsymbol{T}$
$T^{(1)}$
$T^{(2)}$
$\boldsymbol{T}$
$t = (t^{(1)}, t^{(2)})$
C_t
Window (weight) vectors
$(t^{(2)} - 1)T^{(1)}$
$+ t^{(1)}$
$T^{(1)} \times T^{(2)}$
0
1
−0.5
0.5
$\boldsymbol{w}_{\boldsymbol{t}}^{(S)}$
$\boldsymbol{w}_{\boldsymbol{t}}^{(H)}$
$\boldsymbol{w}_{\boldsymbol{t}}^{(V)}$

Figure 5.2: Examples of $\boldsymbol{w}_t^{(S)}$, $\boldsymbol{w}_t^{(H)}$, and $\boldsymbol{w}_t^{(V)}$, 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. (5.11) and (5.12). 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. (5.18), (5.19), and (5.20).

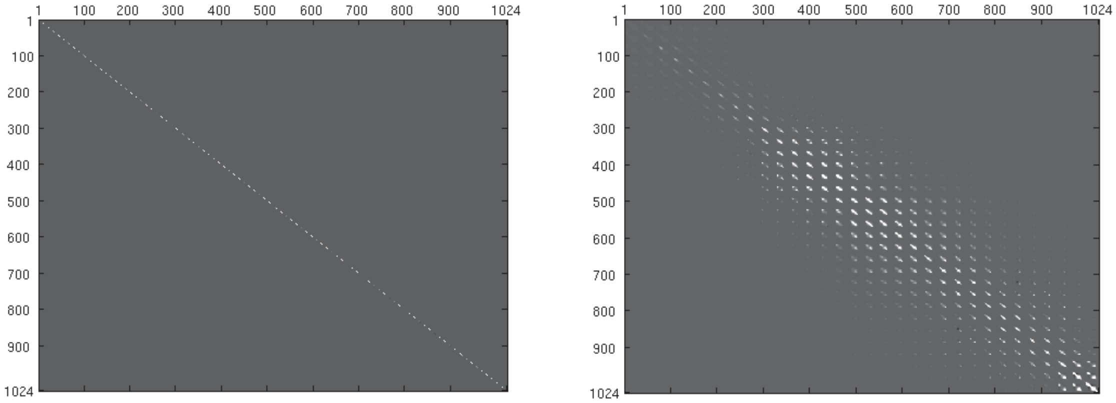
5.2 Model definition

In order to avoid the problem described in the beginning of Chapter 5, that is, the inconsistency between the static and dynamic feature vectors, SL2D-HMMs should be reformulated as the function of $\boldsymbol{C}$ because the original observation is $\boldsymbol{C}$ rather than $\boldsymbol{O}$. Based on the relationship $\boldsymbol{O}$ and $\boldsymbol{C}$ in Eq. (5.15), the definition of the proposed model can be derived.

The output probability $P(\boldsymbol{O} \mid \boldsymbol{S}, \Lambda)$ of SL2D-HMMs is given by

$$P(\boldsymbol{O} \mid \boldsymbol{S}, \Lambda) = \mathcal{N}(\boldsymbol{O} \mid \boldsymbol{\mu}_{\boldsymbol{S}}, \boldsymbol{\Sigma}_{\boldsymbol{S}}) = \prod_{\boldsymbol{t}} \mathcal{N}(\boldsymbol{O}_{\boldsymbol{t}} \mid \boldsymbol{\mu}_{S_{\boldsymbol{t}}}, \boldsymbol{\Sigma}_{S_{\boldsymbol{t}}}), \tag{5.22}$$

where $\mathcal{N}(\cdot \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$, and $\boldsymbol{\mu}_{\boldsymbol{S}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{S}}$ are the "image level" mean vector and covariance matrix given



(a) Covariance matrix of SL2D-HMMs (b) Covariance matrix of SLT2D-HMMs

Figure 5.3: Examples of covariance matrix. (a) shows the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{S}}$ of SL2D-HMMs in Eq. (5.24) and (b) shows the covariance matrix $\boldsymbol{P}_{\boldsymbol{S}}$ of SLT2D-HMMs in Eq. (5.28) 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 32×32. The rows and columns are aligned in raster order of the 2-D lattice (see Fig. 3.3).

state sequences $\boldsymbol{S}$, respectively. They are constructed by concatenating the "state level" mean vectors and covariance matrices in accordance with state sequences $\boldsymbol{S}$:

$$\boldsymbol{\mu}_{\boldsymbol{S}} = \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{S}_{(1,1)}}^\top & \cdots & \boldsymbol{\mu}_{\boldsymbol{S}_t}^\top & \cdots & \boldsymbol{\mu}_{\boldsymbol{S}_{(T^{(1)},T^{(2)})}}^\top \end{bmatrix}^\top, \tag{5.23}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{S}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{S}_{(1,1)}} & & & & \boldsymbol{0} \\ & \ddots & & & \\ & & \boldsymbol{\Sigma}_{\boldsymbol{S}_t} & & \\ & & & \ddots & \\ \boldsymbol{0} & & & & \boldsymbol{\Sigma}_{\boldsymbol{S}_{(T^{(1)},T^{(2)})}} \end{bmatrix}. \tag{5.24}$$

However, Eq. (5.22) becomes an invalid probabilistic distribution over $\boldsymbol{C}$ because the integral of Eq. (5.22) over $\boldsymbol{C}$ is not equal to 1. Namely, Eq. (5.22) is not normalized as the probability distribution of $\boldsymbol{C}$. To yield a valid probability distribution over $\boldsymbol{C}$, Eq. (5.22) can be re-normalized and written as

$$P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) = \frac{1}{Z_{\boldsymbol{S}}} \mathcal{N}(\boldsymbol{W}\boldsymbol{C} \mid \boldsymbol{\mu}_{\boldsymbol{S}}, \boldsymbol{\Sigma}_{\boldsymbol{S}}) = \mathcal{N}(\boldsymbol{C} \mid \overline{\boldsymbol{C}}_{\boldsymbol{S}}, \boldsymbol{P}_{\boldsymbol{S}}), \tag{5.25}$$

$$Z_{\boldsymbol{S}} = \int \mathcal{N}(\boldsymbol{W}\boldsymbol{C} \mid \boldsymbol{\mu}_{\boldsymbol{S}}, \boldsymbol{\Sigma}_{\boldsymbol{S}}) \, \mathrm{d}\boldsymbol{C} \tag{5.26}$$

$$= \frac{\sqrt{(2\pi)^{MT^{(1)}T^{(2)}} |\boldsymbol{P}_{\boldsymbol{S}}|}}{\sqrt{(2\pi)^{3MT^{(1)}T^{(2)}} |\boldsymbol{\Sigma}_{\boldsymbol{S}}|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{S}}^\top \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1} \boldsymbol{\mu}_{\boldsymbol{S}} - \boldsymbol{r}_{\boldsymbol{S}}^\top \boldsymbol{P}_{\boldsymbol{S}} \boldsymbol{r}_{\boldsymbol{S}}\right)\right\}, \tag{5.27}$$

where $Z_{\boldsymbol{S}}$ is a normalization term, and $\overline{\boldsymbol{C}}_{\boldsymbol{S}}$ and $\boldsymbol{P}_{\boldsymbol{S}}$ are the $MT^{(1)}T^{(2)}$ mean vector and the $MT^{(1)}T^{(2)} \times MT^{(1)}T^{(2)}$ covariance matrix, respectively. Also, $\boldsymbol{r}_{\boldsymbol{S}}$, $\overline{\boldsymbol{C}}_{\boldsymbol{S}}$ and $\boldsymbol{P}_{\boldsymbol{S}}$ are given as

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{S}} &= \boldsymbol{W}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{S}}^{-1}, && (5.28) \\ \boldsymbol{r}_{\boldsymbol{S}} &= \boldsymbol{W}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{\mu}_{\boldsymbol{S}}, && (5.29) \\ \overline{\boldsymbol{C}}_{\boldsymbol{S}} &= \boldsymbol{P}_{\boldsymbol{S}}\boldsymbol{r}_{\boldsymbol{S}}. && (5.30) \end{aligned}$$

Please refer Appendix B for detail. Using the above distribution, the joint distribution of static feature vectors $\boldsymbol{C}$ and hidden variables $\boldsymbol{S}$ can be written as:

$$P(\boldsymbol{C}, \boldsymbol{S} \mid \Lambda) = P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) \prod_{m=1,2} P(\boldsymbol{S}^{(m)} \mid \Lambda). \quad (5.31)$$

In the proposed model, the hidden variables are composed of two independent Markov chains, similar to SL2D-HMMs. Therefore, $P(\boldsymbol{S} \mid \Lambda)$ can be factorized into the product of horizontal and vertical state transition probabilities, as shown in Eq. (5.31). By marginalizing $P(\boldsymbol{C}, \boldsymbol{S} \mid \Lambda)$ over all possible state sequences $\boldsymbol{S}$, SL2D-HMMs can be re-defined as follows:

$$\begin{aligned} P(\boldsymbol{C} \mid \Lambda) &= \sum_{\boldsymbol{S}} P(\boldsymbol{C}, \boldsymbol{S} \mid \Lambda) \\ &= \sum_{\boldsymbol{S}} P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) \prod_{m=1,2} P(\boldsymbol{S}^{(m)} \mid \Lambda), && (5.32) \\ P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) &= \frac{1}{Z_{\boldsymbol{S}}} \prod_{t} \mathcal{N}(\boldsymbol{W}\boldsymbol{C}_t \mid \boldsymbol{\mu}_{\boldsymbol{S}_t}, \boldsymbol{\Sigma}_{\boldsymbol{S}_t}) && (5.33) \\ &= \frac{1}{Z_{\boldsymbol{S}}} \mathcal{N}(\boldsymbol{W}\boldsymbol{C} \mid \boldsymbol{\mu}_{\boldsymbol{S}}, \boldsymbol{\Sigma}_{\boldsymbol{S}}) && (5.34) \\ &= \mathcal{N}(\boldsymbol{C} \mid \overline{\boldsymbol{C}}_{\boldsymbol{S}}, \boldsymbol{P}_{\boldsymbol{S}}), && (5.35) \end{aligned}$$

where Λ 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 $\boldsymbol{S}$ in Eq. (5.32) 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 Section 4, a strategy will be described to make this problem computationally tractable. It should be also noted that covariance matrix $\boldsymbol{P}_{\boldsymbol{S}}$ 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 $\boldsymbol{P}_{\boldsymbol{S}}$. As a result, the proposed model can mitigate the limitations of SL2D-HMMs.

Figure 5.3 shows examples of covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{S}}$ of SL2D-HMMs and covariance matrix $\boldsymbol{P}_{\boldsymbol{S}}$ 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 × 32. The detail of the training data and conditions will be described in Section 5.5.1. Note that both the rows and columns are aligned in raster order of the 2-D lattice (see Eq. (3.1) and Fig. 3.3), because the rows of $\boldsymbol{C}$ in Eq. (5.14) 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.

5.3 Relation to other statistical models

It has been discussed in [37] that there exists the relationship between the trajectory HMMs [1] and the product of experts (PoE) [38], especially, product of Gaussian experts (PoG) [39]. 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 [40] 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) [41] is a particular case of MoE where each expert is a normalized Gaussian. According to [37], PoE (PoG) is an efficient way of represent high-dimensional data which simultaneously satisfies many different low-dimensional constraints. In Eq. (5.33), $\mathcal{N}(\boldsymbol{W}\boldsymbol{C}_t \mid \boldsymbol{\mu}_{\boldsymbol{S}_t}, \boldsymbol{\Sigma}_{\boldsymbol{S}_t})$ is an unnormalized Gaussian as a probability distribution of $\boldsymbol{C}_t$. 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 $Z_{\boldsymbol{S}}$ in Eq. (5.33) can be represented in a closed form as Eq. (5.27), without any approximation. Therefore, the output probability $P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda)$ can be evaluated strictly and this helps the great simplification of model training, compered 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 [42] from the interesting discussion of the relationship between 1-D trajectory HMMs and Markov random fields in [37]. The graphical model representation of SLT2D-HMMs can be specified by the window matrix $\boldsymbol{W}$, 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.

5.4 Training algorithm

The parameters of the proposed model can be estimated via the expectation maximization (EM) algorithm [27] 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:

$$Q(\Lambda, \Lambda') = \sum_{\boldsymbol{S}} P(\boldsymbol{S} \mid \boldsymbol{C}, \Lambda) \log P(\boldsymbol{C}, \boldsymbol{S} \mid \Lambda'). \tag{5.36}$$

By maximizing the Q-function with respect to model parameters Λ, the re-estimation formula in the M-step can be easily derived. However, the evaluation of the posterior distribution $P(\boldsymbol{S} \mid \boldsymbol{C}, \Lambda)$ over all possible state sequences $\boldsymbol{S}$ 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:

$$\boldsymbol{S}_{max} = \arg\max_{\boldsymbol{S}} P(\boldsymbol{C}, \boldsymbol{S} \mid \Lambda), \tag{5.37}$$

$$\hat{\Lambda} = \arg\max_{\Lambda} P(\boldsymbol{C}, \boldsymbol{S}_{max} \mid \Lambda). \tag{5.38}$$

However, it is still difficult to solve the problem of Eq. (5.37) because the inter-frame covariance matrix $\boldsymbol{P}_{\boldsymbol{S}}$ is generally full.

5.4.1 Estimation of sub-optimum state sequence

In this section, the Viterbi approximation [26] to solve the maximization problem of Eq. (5.37) is described. This approximation is based on the following relationship

$$\begin{aligned} \boldsymbol{S}_{max} &= \arg\max_{\boldsymbol{S}} P(\boldsymbol{C}, \boldsymbol{S} \mid \Lambda) & (5.39)\\ &= \arg\max_{\boldsymbol{S}} P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) P(\boldsymbol{S} \mid \Lambda) & (5.40)\\ &= \arg\max_{\boldsymbol{S}} \frac{1}{Z_{\boldsymbol{S}}} \mathcal{N}(\boldsymbol{O} \mid \boldsymbol{\mu}_{\boldsymbol{S}}, \boldsymbol{\Sigma}_{\boldsymbol{S}}) P(\boldsymbol{S} \mid \Lambda) & (5.41)\\ &\approx \arg\max_{\boldsymbol{S}} \mathcal{N}(\boldsymbol{O} \mid \boldsymbol{\mu}_{\boldsymbol{S}}, \boldsymbol{\Sigma}_{\boldsymbol{S}}) P(\boldsymbol{S} \mid \Lambda), & (5.42) \end{aligned}$$

where the Viterbi approximation is applied in Eq. (5.42). Let $\boldsymbol{S}_{sub} = \left(\boldsymbol{S}^{(1)}_{sub}, \boldsymbol{S}^{(2)}_{sub}\right)$ be a sub-optimum state sequence for SLT2D-HMMs. In order to obtain $\boldsymbol{S}_{sub}$ from all possible state sequence, following approximation strategy was adopted in this paper:

Step 1 Initialize $\boldsymbol{S}_{sub}$ with the Viterbi state sequence $\boldsymbol{S}_{vit} = \left(\boldsymbol{S}^{(1)}_{vit}, \boldsymbol{S}^{(2)}_{vit}\right)$ of SL2D-HMMs.

Step 2 Add small variations on each boundary of $\boldsymbol{S}^{(1)}_{sub}$ and $\boldsymbol{S}^{(2)}_{sub}$ and collect resulting state sequences as candidates. In this paper, the small variations were shift of ±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.

5.4.2 Estimation of model parameters

In this section, the maximization problem of Eq. (5.38) is described. The problem is equivalent to maximizing the log-likelihood

$$\begin{aligned} &\log P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) \\ = \; & -\frac{1}{2}\left\{MT^{(1)}T^{(2)}\log(2\pi) - \log|\boldsymbol{R}_S| + \boldsymbol{C}^\top \boldsymbol{R}_S \boldsymbol{C} + \boldsymbol{r}_S^\top \boldsymbol{P}_S \boldsymbol{r}_S - 2\boldsymbol{r}_S^\top \boldsymbol{C}\right\} \end{aligned} \tag{5.43}$$

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

$$\boldsymbol{m} = \left[\begin{array}{ccccc} \boldsymbol{\mu}^\top_{(1,1)} & \cdots & \boldsymbol{\mu}^\top_{\boldsymbol{k}} & \cdots & \boldsymbol{\mu}^\top_{(K^{(1)},K^{(2)})} \end{array}\right]^\top, \tag{5.44}$$

$$\boldsymbol{\phi} = \left[\begin{array}{ccccc} \boldsymbol{\Sigma}^{-1}_{(1,1)} & \cdots & \boldsymbol{\Sigma}^{-1}_{\boldsymbol{k}} & \cdots & \boldsymbol{\Sigma}^{-1}_{(K^{(1)},K^{(2)})} \end{array}\right]^\top. \tag{5.45}$$

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

$$\boldsymbol{\mu}_S = \boldsymbol{F}_S \boldsymbol{m}, \quad \boldsymbol{\Sigma}_S^{-1} = \mathrm{diag}[\boldsymbol{F}_S \boldsymbol{\phi}]. \tag{5.46}$$

By using $\boldsymbol{F}_S$, Eqs. (5.28) and (5.29) can be written as

$$\boldsymbol{R}_S = \boldsymbol{W}^\top \cdot \mathrm{diag}[\boldsymbol{F}_S \boldsymbol{\phi}] \cdot \boldsymbol{W} = \boldsymbol{P}_S^{-1}, \tag{5.47}$$

$$\boldsymbol{r}_S = \boldsymbol{W}^\top \cdot \mathrm{diag}[\boldsymbol{F}_S \boldsymbol{\phi}] \cdot \boldsymbol{F}_S \boldsymbol{m}. \tag{5.48}$$

According to (5.47) and (5.48), Eq. (5.43) can be re-written as

$$\begin{aligned}\log P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) &= -\frac{1}{2}\Big\{MT^{(1)}T^{(2)} \log(2\pi) - \log \left|\boldsymbol{W}^\top \mathrm{diag}[\boldsymbol{F_S}\boldsymbol{\phi}]\boldsymbol{W}\right| \\ &\quad +\boldsymbol{C}^\top \boldsymbol{W}^\top \mathrm{diag}[\boldsymbol{F_S}\boldsymbol{\phi}]\boldsymbol{W}\boldsymbol{C} \\ &\quad +\boldsymbol{m}^\top \boldsymbol{F}_{\boldsymbol{S}}^\top(\mathrm{diag}[\boldsymbol{F_S}\boldsymbol{\phi}])\boldsymbol{W}^\top \boldsymbol{P_S}\boldsymbol{W}(\mathrm{diag}[\boldsymbol{F_S}\boldsymbol{\phi}])\boldsymbol{F_S}\boldsymbol{m} \\ &\quad -2\boldsymbol{m}^\top \boldsymbol{F}_{\boldsymbol{S}}^\top(\mathrm{diag}[\boldsymbol{F_S}\boldsymbol{\phi}])\boldsymbol{W}^\top \boldsymbol{C}\Big\}. \qquad (5.49)\end{aligned}$$

Therefore, a partial derivative of Eq. (5.43) with respect to $\boldsymbol{m}$ and $\boldsymbol{\phi}$ can be written as

$$\frac{\partial \log P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda)}{\partial \boldsymbol{m}} = \boldsymbol{F}_{\boldsymbol{S}}^\top \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1} \boldsymbol{W}\left(\boldsymbol{C} - \overline{\boldsymbol{C}}_{\boldsymbol{S}}\right), \qquad (5.50)$$

$$\frac{\partial \log P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda)}{\partial \boldsymbol{\phi}} = \frac{1}{2}\boldsymbol{F}_{\boldsymbol{S}}^\top \mathrm{diag}^{-1}\left[\boldsymbol{W}\boldsymbol{G_S}\boldsymbol{W}^\top + 2\boldsymbol{\mu_S}(\boldsymbol{C} - \overline{\boldsymbol{C}}_{\boldsymbol{S}})^\top \boldsymbol{W}^\top\right], \qquad (5.51)$$

where $\boldsymbol{G_S} = \boldsymbol{P_S} + \overline{\boldsymbol{C}}_{\boldsymbol{S}}\overline{\boldsymbol{C}}_{\boldsymbol{S}}^\top - \boldsymbol{C}\boldsymbol{C}^\top$ and diag^{-1} denotes the extraction of only diagonal elements from a square matrix. By setting Eq. (5.50) equals to $\boldsymbol{0}_{3MK^{(1)}K^{(2)}}$ and solving the resultant linear equation, the following re-estimation formula for the supervector $\boldsymbol{m}$ maximizing Eq. (5.43) can be obtained:

$$\hat{\boldsymbol{m}} = \boldsymbol{A}^{-1}\boldsymbol{b}, \qquad (5.52)$$

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

$$\boldsymbol{A} = \boldsymbol{G}_{\boldsymbol{S}}^\top \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1} \boldsymbol{W}\boldsymbol{P_S}\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{G_S}, \qquad (5.53)$$

$$\boldsymbol{b} = \boldsymbol{G}_{\boldsymbol{S}}^\top \boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{W}\boldsymbol{C}. \qquad (5.54)$$

Please refer Appendix C for detail of the above formula. For maximizing Eq. (5.43) with respect to $\boldsymbol{\phi}$, a gradient method can be applied using its first derivative of Eq. (5.51).

5.4.3 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 $\boldsymbol{m}$ and $\boldsymbol{\phi}$.

Step 3 Search sub-optimal state sequences in accordance with the procedure as summarized in Section 5.4.1.

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

5.5 Experiments

5.5.1 Experimental conditions

To demonstrate the effectiveness of the proposed model, experiments on modeling faces from the XM2VTS database [35] were conducted. The face images were extracted from the original images (720 × 576 pixels and transformed into gray-scale) and then subsampled to 16×16 and 32×32 pixels. The images of 16×16 pixels were used for image recognition experiments and the images of 32×32 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 40×20 pixels from the center and the range of sizes $500 \times 500 \sim 600 \times 600$ with a fixed aspect ratio.

Figure 5.4 shows the examples of two datasets where the size of face image is 16×16. 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 $\boldsymbol{O}$ 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 $\boldsymbol{C}$. In the case of (ii), an observation vector $\boldsymbol{O}_t$ can be constructed as

$$\boldsymbol{O}_t = \left[\Delta^{(S)}\boldsymbol{C}_t^\top, \Delta^{(H)}\boldsymbol{C}_t^\top, \Delta^{(V)}\boldsymbol{C}_t^\top, \Delta^{(D_1)}\boldsymbol{C}_t^\top, \Delta^{(D_2)}\boldsymbol{C}_t^\top\right]^\top, \tag{5.55}$$

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

$$\Delta^{(D_1)}\boldsymbol{C}_t = \sum_{\tau=-L_-^{(D_1)}}^{L_+^{(D_1)}} w^{(D_1)}(\tau)\boldsymbol{C}_{(t^{(1)}-\tau, t^{(2)}+\tau)}, \tag{5.56}$$

$$\Delta^{(D_2)}\boldsymbol{C}_t = \sum_{\tau=-L_-^{(D_2)}}^{L_+^{(D_2)}} w^{(D_2)}(\tau)\boldsymbol{C}_{(t^{(1)}+\tau, t^{(2)}+\tau)}. \tag{5.57}$$

For each case, the corresponding window matrix $\boldsymbol{W}$ was designed to satisfy Eq. (5.15). In the case of (i),

$$L_+^{(H)} = L_-^{(H)} = L_+^{(V)} = L_-^{(V)} = 1.0, \tag{5.58}$$

$$w^{(H)}(-1) = w^{(V)}(-1) = -0.5, \tag{5.59}$$

$$w^{(H)}(0) = w^{(V)}(0) = 0.0, \tag{5.60}$$

$$w^{(H)}(1) = w^{(V)}(1) = 0.5. \tag{5.61}$$

Additionally, in the case of (ii),

$$L_{+}^{(D_1)} = L_{-}^{(D_1)} = L_{+}^{(D_2)} = L_{-}^{(D_2)} = 1.0, \tag{5.62}$$

$$w^{(D_1)}(-1) = w^{(D_2)}(-1) = -0.5, \tag{5.63}$$

$$w^{(D_1)}(0) = w^{(D_2)}(0) = 0.0, \tag{5.64}$$

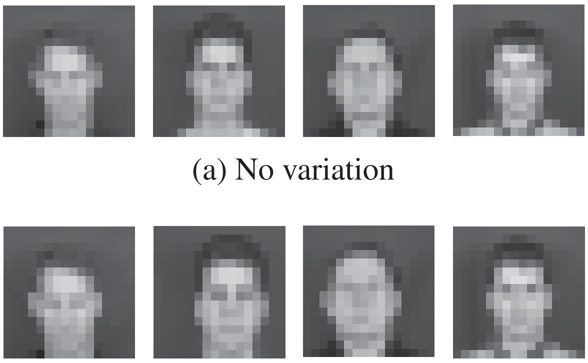
$$w^{(D_1)}(1) = w^{(D_2)}(1) = 0.5. \tag{5.65}$$

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, the SL2D-HMMs were extended by integrating with a linear feature extraction such as probabilistic PCA or factor analyzers [10]. 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 [10] for improving recognition performance.

The model parameters of SLT2D-HMMs were estimated in accordance with the training procedure as summarized in Section 5.4. To make the concatenated covariance matrix $\boldsymbol{\phi}$ be positive, $\log(\boldsymbol{\phi})$ was used in optimizing $\boldsymbol{\phi}$, where $\log(\cdot)$ denotes elementwise logarithm operator. The Rprop method [43], a first order gradient-based optimization method, was adopted for optimizing $\log(\boldsymbol{\phi})$ in this paper.

5.5.2 Face recognition experiments

Face recognition experiments on the XM2VTS database were conducted. We prepared eight images (two images × 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 16×16 and they were modeled by SL2D-HMMs and SLT2D-HMMs with 4×4, 6×6, 8×8, 10×10, and 12×12 states. Fig. 5.5 shows recognition rates of SL2D-HMMs and SLT2D-HMMs.



(a) No variation

(b) Size and location variations

Figure 5.4: Examples of training data; with no variation (a) and with variations of size and location (b). The size of face image is 16×16.

Fig. 5.5(a) and (b) show the results on “dataset1” and “dataset2,” in which 1st order horizontal and vertical dynamic features were applied, respectively. Fig. 5.5(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. 5.5(b) were higher than those in Fig. 5.5(a) as a whole. Especially, in Fig. 5.5(a), the recognition rate of 51.5% was obtained at 8×8 states of “ParamUpdate,” while, in Fig. 5.5(b), the highest recognition rate of 54.3% was obtained at the same states of “ParamUpdate.” Similar tendency could be observed from Fig. 5.5(c) and Fig. 5.5(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 the SLT2D-HMMs. After

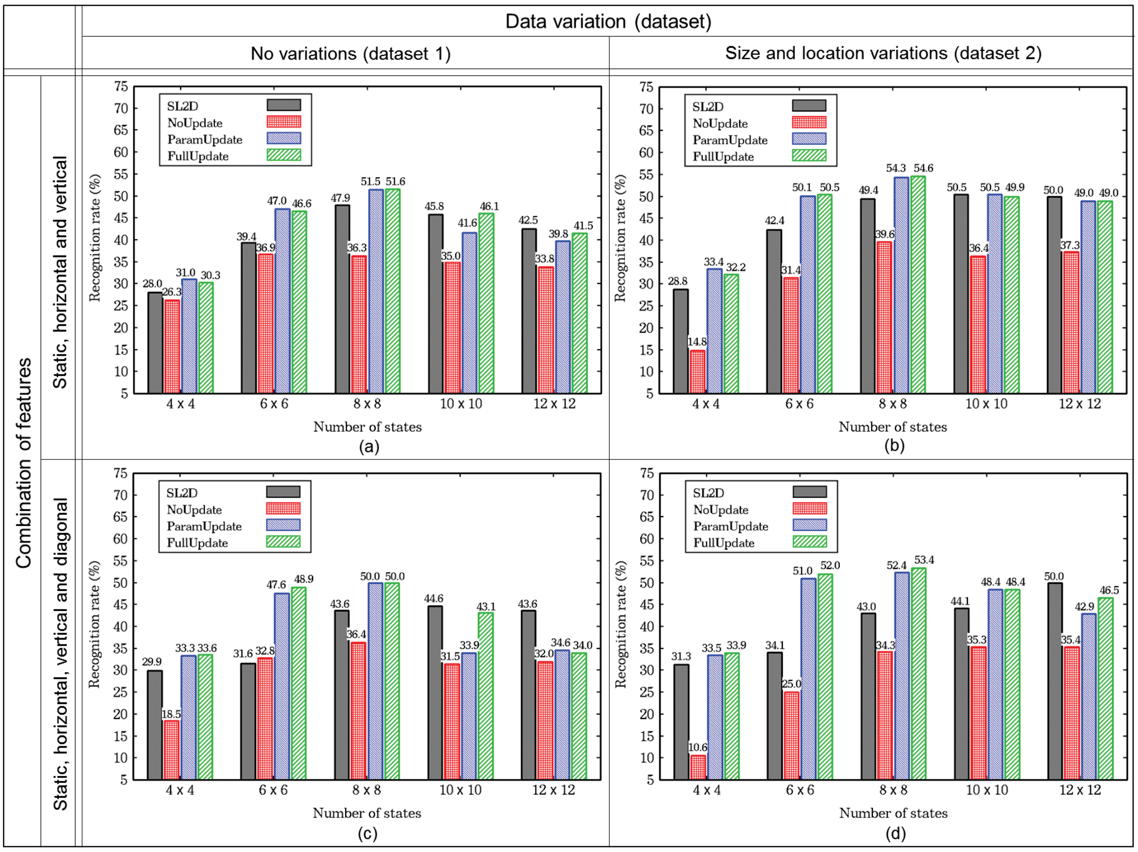


Figure 5.5: Recognition rates of SL2D-HMMs and SLT2D-HMMs. Two figures on the top , (a) and (b) show the results on “dataset1” and “dataset2,” in which 1st order horizontal and vertical dynamic features were applied, respectively. On the other hand, two figures on the bottom, (c) and (d) show the results 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 × 16.

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 Section 5.4.1 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. However, much more computational time will be required in this strategy.

From Fig. 5.5(a) and (c), it can be seen that the recognition rates in Fig. 5.5(c) were slightly lower than those in Fig. 5.5(a). In particular, the highest recognition rate of 50.0% at 8 × 8 states of “FullUpdate” in Fig. 5.5(c) was lower than that of 51.6% at the same states of “FullUpdate” in Fig. 5.5(a). This is partly because the model over-fitted to the training data with size and location variations.

5.5.3 State alignment experiments

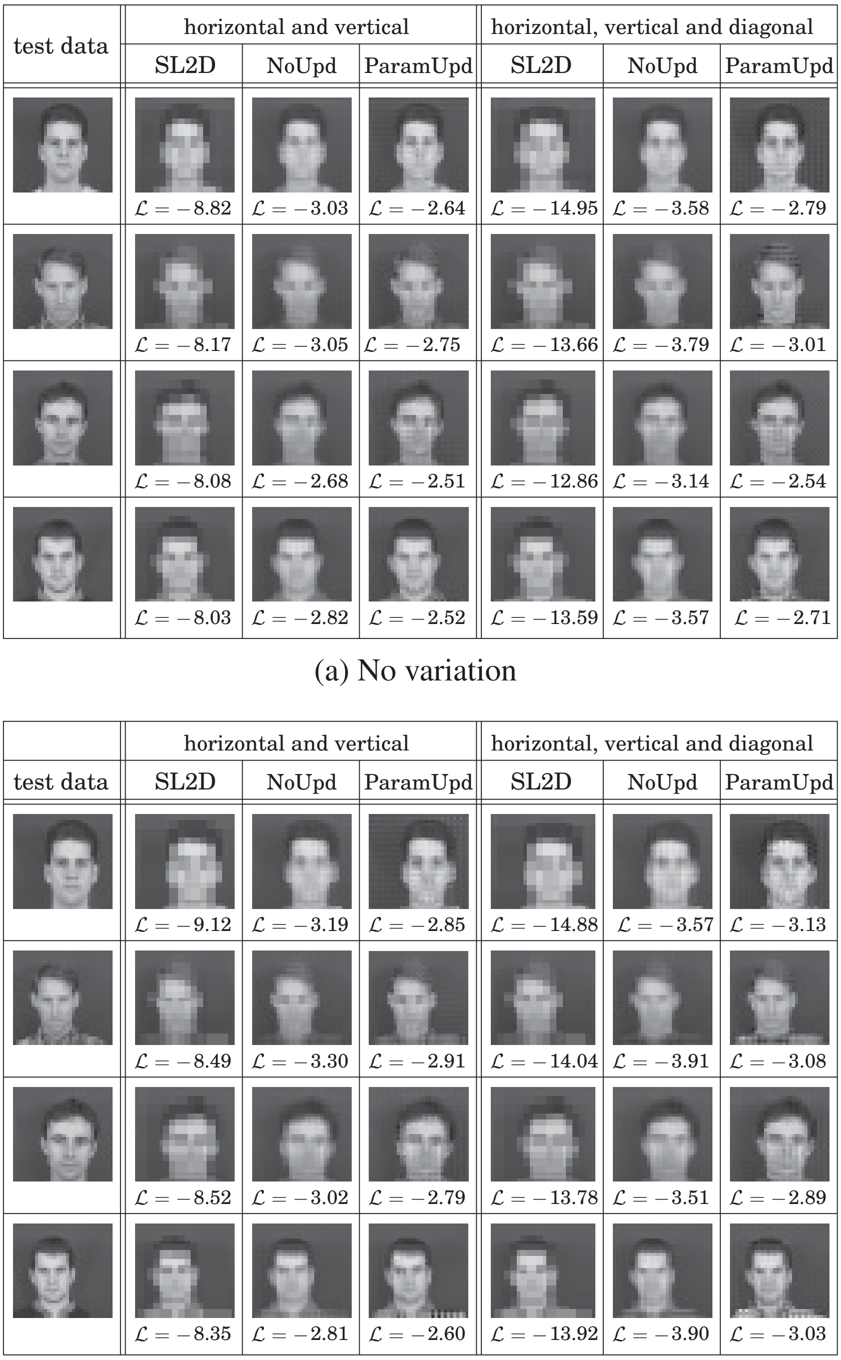
To demonstrate the advantageous property of SLT2D-HMMs for image recognition, an state alignment experiment was conducted on “dataset1” and “dataset2,” where the size of the face images was 32 × 32 and the number of HMM-states was 16 × 16. Figure 5.6 shows the test image and its state alignments of SL2D-HMMs 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. 5.6(a), it can be seen that a rectangular state alignment was obtained by using the 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 $\overline{\boldsymbol{C}}_{\boldsymbol{S}}$ 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., 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. 5.6(a). Moreover, it can also be seen from “SL2D” of Fig. 5.6(b) that SL2D-HMMs could deal with size and location variation by changing the each state duration. From “NoUpd” and “ParamUpd” of Fig. 5.6(b), this property also holds true in SLT2D-HMMs. These results also explain the improvement of the recognition performance.

From both Fig. 5.6(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 $\boldsymbol{O}$ in SL2D-HMMs were composed of the static and dynamic features, while the observation vectors in SLT2D-HMMs were the only static features $\boldsymbol{C}$. 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 Fig. 5.6(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.

5.6 Summary

In this chapter, a novel statistical model based on 2-D HMMs for image recognition has been derived. It has been known that SL2D-HMMs have the 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 the SL2D-HMMs.



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

	horizontal and vertical			horizontal, vertical and diagonal		
test data	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

Figure 5.6: 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 × 32 and the number of states is 16 × 16. The $\mathcal{L}$ means the estimated log-likelihood per pixel to test data.

Chapter 6

Conclusions

6.1 Summary

The present dissertation described novel statistical models based on separable lattice 2-D HMMs for image recognition.

Basic theories and fundamental algorithms of the HMM were reviewed in Chapter 2 and Chapter 3 described the model definition and the training algorithm of separable lattice 2-D HMMs. SL2D-HMMs have the composite structure of multiple hidden state sequences which interact to model the observation on a lattice. SL2D-HMMs perform an elastic matching in both horizontal and vertical directions; this makes it possible to model not only invariance to the size and location of an object but also nonlinear warping in each dimension. Although the training algorithm of SL2D-HMMs based on EM algorithm can be derived, the complexity of the exact E-step become an exponential order and therefore, it is computationally intractable. To derive a feasible problem, the variational EM algorithm and variational DAEM algorithm can be derived. In the both algorithms, the complexity of the E-step can be reduced to a polynomial order.

In Chapter 4, an extension of SL2D-HMMs for rotational variations was proposed. Although the proposed model has potential to perform an continuous elastic matching beyond rotational variations, the topology and the shift amounts are constrained to a special form which is expected to represent the continuous rotational variations in this dissertation. For model training, the variational EM algorithm can also be applied to the proposed model. In face recognition experiments on the XM2VTS database, the proposed model achieved better results to the images than the conventional SL2D-HMMs. Moreover, the state alignments shows that the proposed model can normalize not only size and location variations but also rotational variations.

In Chapter 5, the separable lattice 2-D trajectory HMMs (SLT2D-HMMs) were derived. It has been known that SL2D-HMMs have the shortcomings which are inherited from standard HMMs. To overcome these shortcomings of SL2D-HMMs, the present dissertation derives 2-D trajectory HMMs by reformulating the likelihood of SL2D-HMMs with imposing explicit relationships between static and dynamic features. The proposed model can efficiently capture dependencies between adjacent observations without any additional model parameters. The effectiveness of the proposed model was evaluated in face recognition experiments on XM2VTS database. The proposed model achieved better results than the SL2D-HMMs.

6.2 Future work

For an extension of SL2D-HMMs to deal with rotational variations, integration with a linear feature extraction as [10] to improve recognition performance will be future work. For SLT2D-HMMs, we are to plan to append not only 1st order dynamic features, but also more higher order dynamic features. Extending SLT2D-HMMs for rotational variations will be future work. Moreover, implementing more precise search algorithms such as the delayed decision Viterbi algorithm [1] will be also future work. For both proposed models, applying the Bayesian criterion [44] is to be investigated. Conducting experiments on various image recognition tasks and with other statistical methods, e.g., support vector machines [45], neural networks [46], and MRFs [47] will also be future work to evaluate the effectiveness of the proposed model over these methods.

Bibliography

[1] 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*. PhD thesis, Nagoya Institute of Technology, 2006.

[2] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE*, Vol. 77, pp. 257–285, 1989.

[3] F. Samaria and F. Fallside. Face identification and feature extraction using hidden markov models. In *Image Processing: Theory and Applications*, pp. 295–298. Elsevier, 1993.

[4] S. Kuo and O.E. Agazzi. Keyword Spotting in Poorly Printed Documents Using Pseudo 2-D Hidden Markov Models. *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 16, pp. 842–848, 1994.

[5] A. 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.

[6] H. Othman and T. Aboulnasr. A Simplifed Second-Order HMM with Application to Face Recognition. *International Symposium on Circuits and Systems*, Vol. 2, pp. 161–164, 2001.

[7] J.-T. Chien and C.-P. Liao. Maximum Confidence Hidden Markov Modeling for Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, pp. 606–616, 2008.

[8] J. Li, A. Najmi, and R. M. Gray. Image Classification by a Two-Dimensional Hidden Markov Model. *IEEE Transations on Signal Processing*, Vol. 48, No. 2.

[9] D. Kurata, Y. Nankaku, K. Tokuda, T. Kitamura, and Z. Ghahramani. Face recognitinon based sepalable lattice HMMs. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 737–740, 2006.

[10] 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.

[11] A. Tamamori, Y. Nankaku, and K. Tokuda. An Extension of Separable Lattice 2-D HMMs for Rotational Data Variations. *IEICE Transactions on Information & Systems*, Vol. E95-D, No. 8, pp. 2074–2083, 2012.

[12] 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.

[13] 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.

[14] S. Uchida and H. Sakoe. An Approximation Algorithm for Two-Dimensional Warping. *IEICE Transactions on Information & Systems*, Vol. E83-D, No. 1, pp. 109–111, 2000.

[15] S. Uchida and H. Sakoe. Piecewise linear two-dimensional warping. *Systems and Computers in Japan*, Vol. 32, No. 12, pp. 1–9, 2001.

[16] 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.

[17] Y. Yaguchi, K. Iseki, N. T. Viet, and R. Oka. Full Pixel Matching between Images for Non-linear Registration of Objects. *IPSJ Transactions on Computer Vision and Applications*, Vol. 2, pp. 1–14, 2010.

[18] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, Vol. 32, pp. 183–233, 1999.

[19] S. Nakagawa. A survey on automatic speech recognition. *IEICE Transactions Information and Systems* , Vol. E85-D (3), pp. 465–486, 2002.

[20] S. Furui. Speaker independent isolated word recognition using dynamic features of speech spetrum. *IEEE Transactions Acoustics, Speech, and Signal Processing*, Vol. 34, pp. 52–59, 1986.

[21] K. Kumaki. *Face Recognition based on Extended Separable Lattice HMMs*. Bachelor thesis, Nagoya Insititute of Technology, 2010.

[22] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov models mor speech recognition*. Edinburgh University Press, 1990.

[23] L. Rabiner and B.H. Juang. *Fundamentals of speech recognition*. Prentice-Hall, 1993.

[24] S. Young, G. Evermann, M.J.F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.4)*. Cambridge University, 2005.

[25] L. Rabiner, B.H. Juang, S.E. Levinson, and M.M. Sondhi. Recognition of isolated digits using hidden Markov models with continuous mixture densities. *AT&T Technical Journal*, Vol. 64, No. 6, pp. 1211–1234, 1985.

[26] A. Viterbi. Error bounds for convolutinal codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, Vol. 13, pp. 260–269, 1967.

[27] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from imcomplete data via the EM algorithm. *Journal of Royal Statistics Sosiety*, Vol. 39, pp. 1–38, 1997.

[28] B.H. Juang. Maximum likelihood estimation for mixture of multivariate stochastic observations of Markov c hains. *AT&T Technical Journal*, Vol. 64, No. 6, pp. 1235–1249, 1985.

[29] M. Turk and A. Pentland. Face recognition using eigenfaces. *IEEE Computer Society Conference*, pp. 586–591, 1991.

[30] E. Oja. *Subspace Method for Pattern Recognition*. Research Studies Press, 1983.

[31] E. Levin and R. Pieraccini. Dynamic Planar Warping for Optical Character Recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 3, pp. 149–152, 1992.

[32] J. Li, A. Najmi, and M. Gray. Image Classification by a Two-Dimensional Hidden Markov Model. *IEEE Trans. Signal Processing*, Vol. 48, No. 2, pp. 517–533, 2000.

[33] S. Müller, S. Eickeler, and G. Rigoll. Pseudo 3-D HMMs for Image Sequence Recognition. *IEEE International Conference on Image Processing (ICIP)*, pp. 237–241, 1999.

[34] N. Ueda and R. Nakano. Deterministic annealing em algorithm. *Neural Networks*, Vol. 11, No. 2, pp. 271–282, 1998.

[35] K. Messer, J. Mates, J. Kitter, J. Luettin, and G. Maitre. XM2VTS: The Extended M2VTS Database. *Audio and Video-Based Biometric Person Authentication*, pp. 72–77, 1999.

[36] C.-H. Lee and E. Giachin. Speaker independent isolated word recognition using dynamic features of speech spetrum. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 161–164, 1991.

[37] H. Zen, Y. Nankaku, K. Tokuda, and T. Kitamura. Product of Experts for Statistical Parametric Speech Synthesis. *IEEE Transaction on audio, speech and language processing*, Vol. 20, No. 3, pp. 153–173, March 2012.

[38] G. Hinton. Product of experts. *International Conference on Artificial Neural Networks (ICANN)*, Vol. 1, pp. 1–6, 1999.

[39] M. Gales and S. Airey. Product of Gaussians for speech recognition. Technical report, CUED/F-INFENG/TR.458, 2003.

[40] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, Vol. 3, No. 1, pp. 79–87, 1991.

[41] D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using Gaussian Mixture speaker models. *IEEE Tranastions on Speech and Audio Processing*, Vol. 3, No. 1, pp. 72–83, 1995.

[42] H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapamn & Hall/CRC, 2005.

[43] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The Rprop algorithm. *Proceedings of IEEE ICNN*, pp. 586–591, 1993.

[44] K. Sawada, A. Tamamori, Y. Nankaku, and K. Tokuda. Face Recognitinon based Sepalable Lattice 2-D HMMs using Variational Bayesian method. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2205–2208, 2012.

[45] O. Deniz, M. Castrillon, and M. Hernandez. Face recognition using independent component analysis and support vector machines. *Pattern Recognition Letters*, Vol. 24, No. 13, pp. 2153–2157, 2003.

[46] M. J. Er, S. Wu, J. Lu, and H. L. Toh. Face recognition with radial basis funtion (RBF) neural networks. *IEEE Transactions on Neural Networks*, Vol. 13, No. 3, pp. 697–710, 2002.

[47] J. Cal and Z.-Q. Liu. Pattern recognition using Markov Random Field models. *Pattern Recognition*, Vol. 35, No. 3, pp. 725–733, 2002.

List of Publications

Journal dissertations

[1] **Akira Tamamori**, Yoshihiko Nankaku, and Keiichi Tokuda, “Image recognition based on separable lattice trajectory 2-D HMMs,” *IEICE Transactions on Information & Systems*, vol. E97-D, no. 7, pp. 1842–1854, Jul. 2014.

[2] **Akira Tamamori**, Yoshihiko Nankaku, and Keiichi Tokuda, “An Extension of Separable Lattice 2-D HMMs for Rotational Data Variations,” *IEICE Transactions on Information & Systems*, vol. E95-D, no. 8, pp. 2074–2083, Aug. 2012.

International conference proceedings

[3] **Akira Tamamori**, Yoshihiko Nankaku, and Keiichi Tokuda, “Image Recognition Based On Separable Lattice Trajectory 2-D HMMs,” *Proceedings of ICASSP 2013*, pp. 3467–3471, May 2013.

[4] Kei Sawada, **Akira Tamamori**, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Face Recognition Based On Separable Lattice 2-D HMMs Using Variational Bayesian Method,” *Proceedings of ICASSP 2012* pp. 2205–2208, Mar. 2012.

[5] **Akira Tamamori**, Yoshihiko Nankaku, Keiichi Tokuda, “An Extension Of Separable Lattice 2-D HMMs For Rotational Data Variations”, *Proceedings of ICASSP 2010*, pp. 2206–2209, Mar. 2010.

[6] Yoshiaki Takahashi, **Akira Tamamori**, Yoshihiko Nankaku, Keiichi Tokuda, “Face

Recognition Based On Separable Lattice 2-D HMMs With State Duration Modeling," *Proceedings of ICASSP 2010*, pp. 2162–2165, Mar. 2010.

Technical reports

[**8**] Kei Sawada, **Akira Tamamori**, Yoshihiko Nankaku, Keiichi Tokuda, "Face Recognition Based On Separable Lattice 2-D HMMs With Variational Bayesian Method," *Technical Report of IEICE*, vol .111, no .317, PRMU2011-120, pp. 125–130, Nov. 2011.

[**9**] **Akira Tamamori**, Yoshihiko Nankaku, Keiichi Tokuda, "Face recognition based on separable lattice 2-D HMM considering rotational variations," *Technical Report of IEICE*, vol. 108, no. 484, PRMU2008-263, pp. 159–164, Mar. 2009.

[**10**] Yoshiaki Takahashi, **Akira Tamamori**, Yoshihiko Nankaku, Keiichi Tokuda, "Face Recognition Based On Separable Lattice 2-D HMM With State Duration Modeling," *Techinical Report of IEICE*, vol. 108, no. 484, PRMU2008-262, pp. 153-5158, Mar. 2009.

Domestic conference proceedings

[**11**] Kei Sawada, **Akira Tamamori**, Yoshihiko Nankaku, Keiichi Tokuda, "A Training Algorithm Based On Variational Bayesian Method Using Deterministic Annealing Process For Separable Lattice 2-D Hmms", *Proceedings of IPSJ*, vol. 2, pp. 409–410, Mar. 2012.

Appendix A

Derivation of training algorithm for an extension of separable lattice 2-D HMMs for rotational variations

A.1 Derivation of approximated posterior distributions

For this derivation, following abbreviations are adopted.

$$\begin{aligned}
I_Q(\boldsymbol{S}^{(m)}) &= -\sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) \ln Q(\boldsymbol{S}^{(m)}) && \text{(A.1)}\\
I_Q(\boldsymbol{d}^{(m)}) &= -\sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{d}^{(m)}) \ln Q(\boldsymbol{d}^{(m)}) && \text{(A.2)}\\
J_P(\boldsymbol{S}^{(m)}) &= \sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) \ln P(\boldsymbol{S}^{(m)}|\Lambda_S^{(m)}) && \text{(A.3)}\\
J_P(\boldsymbol{S}^{(1)}, \boldsymbol{S}^{(2)}, \boldsymbol{d}^{(1)}, \boldsymbol{d}^{(2)}) &= \sum_{\boldsymbol{S}^{(1)}} \sum_{\boldsymbol{S}^{(2)}} \sum_{\boldsymbol{d}^{(1)}} \sum_{\boldsymbol{d}^{(2)}} Q(\boldsymbol{S}^{(1)}) Q(\boldsymbol{S}^{(2)}) Q(\boldsymbol{d}^{(1)}) Q(\boldsymbol{d}^{(2)}) \\
&\quad \cdot \ \ln P(\boldsymbol{O}|\boldsymbol{S}^{(1)}, \boldsymbol{S}^{(2)}, \boldsymbol{d}^{(1)}, \boldsymbol{d}^{(2)}, \Lambda) && \text{(A.4)}
\end{aligned}$$

Using above abbreviations, the lower bound $\mathcal{F}$ in Eq. (4.11) can be re-written as follows:

$$
\begin{aligned}
\mathcal{F}(Q,\Lambda) &= \sum_{\boldsymbol{S}}\sum_{\boldsymbol{d}} Q(\boldsymbol{S},\boldsymbol{d})\ln P(\boldsymbol{O},\boldsymbol{S},\boldsymbol{d}|\Lambda) - \sum_{\boldsymbol{S}}\sum_{\boldsymbol{d}} Q(\boldsymbol{S},\boldsymbol{d})\ln Q(\boldsymbol{S},\boldsymbol{d}) \\
&= \sum_{\boldsymbol{S}}\sum_{\boldsymbol{d}} Q(\boldsymbol{S},\boldsymbol{d})\ln P(\boldsymbol{O}|\boldsymbol{S},\boldsymbol{d},\Lambda) + \sum_{\boldsymbol{S}}\sum_{\boldsymbol{d}} Q(\boldsymbol{S},\boldsymbol{d})\ln P(\boldsymbol{S},\boldsymbol{d}|\Lambda) \\
&\quad - \sum_{\boldsymbol{S}}\sum_{\boldsymbol{d}} Q(\boldsymbol{S},\boldsymbol{d})\ln Q(\boldsymbol{S},\boldsymbol{d}) \\
&= \sum_{\boldsymbol{S}}\sum_{\boldsymbol{d}} Q(\boldsymbol{S})Q(\boldsymbol{d})\ln P(\boldsymbol{O}|\boldsymbol{S},\boldsymbol{d},\Lambda) + \sum_{\boldsymbol{S}} Q(\boldsymbol{S})\ln P(\boldsymbol{S}|\Lambda) \\
&\quad + \sum_{\boldsymbol{d}} Q(\boldsymbol{d})\ln P(\boldsymbol{d}|\Lambda) - \sum_{\boldsymbol{S}}\sum_{\boldsymbol{d}} Q(\boldsymbol{S})Q(\boldsymbol{d})\ln Q(\boldsymbol{S})Q(\boldsymbol{d}) \qquad \text{(A.5)} \\
&= J_P(\boldsymbol{S}^{(1)},\boldsymbol{S}^{(2)},\boldsymbol{d}^{(1)},\boldsymbol{d}^{(2)}) \\
&\quad + \sum_{m=1,2}\left\{J_P(\boldsymbol{S}^{(m)}) + J_P(\boldsymbol{d}^{(m)}) + I_Q(\boldsymbol{S}^{(m)}) + I_Q(\boldsymbol{d}^{(m)})\right\} \qquad \text{(A.6)}
\end{aligned}
$$

The optimal variable function $Q(\boldsymbol{S},\boldsymbol{d})$ to maximize the functional $\mathcal{F}(Q,\Lambda)$ are constrained under the following equations:

$$
\sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) = 1, \quad \sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{d}^{(m)}) = 1 \qquad \text{(A.7)}
$$

From the method of Language multiplier, it is enough to maximize the following $\mathcal{G}$:

$$
\begin{aligned}
\mathcal{G} &= \mathcal{F}(Q,\Lambda) - \sum_{m=1,2}\lambda_{S,m}\left(\sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) - 1\right) - \sum_{m=1,2}\lambda_{d,m}\left(\sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{d}^{(m)}) - 1\right) \\
&= J_P(\boldsymbol{S}^{(1)},\boldsymbol{S}^{(2)},\boldsymbol{d}^{(1)},\boldsymbol{d}^{(2)}) \\
&\quad + \sum_{m=1,2}\left\{J_P(\boldsymbol{S}^{(m)}) + \lambda_{S,m}\sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) + J_P(\boldsymbol{d}^{(m)}) + \lambda_{d,m}\sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{d}^{(m)})\right\} \\
&\quad + \sum_{m=1,2}\left\{I_Q(\boldsymbol{S}^{(m)}) + I_Q(\boldsymbol{d}^{(m)})\right\} + \lambda_{S,1} + \lambda_{S,2} + \lambda_{d,1} + \lambda_{d,2} \qquad \text{(A.8)}
\end{aligned}
$$

where $\lambda_{S,m}$, $\lambda_{d,m}$ are Lagrange multiplier. To optimize $\mathcal{G}$, it is needed to solve the Euler-Lagrange equations. Since $\mathcal{G}$ contains no differential terms of Q, it is enough to solve following equations for each variable functions $Q(\boldsymbol{S}^{(m)})$, $Q(\boldsymbol{d}^{(m)})$:

$$
\begin{aligned}
\frac{\partial\mathcal{F}}{\partial Q(\boldsymbol{S}^{(m)})} + \lambda_{S,m}^{(m)} &= 0, \qquad \text{(A.9)} \\
\frac{\partial\mathcal{F}}{\partial Q(\boldsymbol{d}^{(m)})} + \lambda_{d,m}^{(m)} &= 0. \qquad \text{(A.10)}
\end{aligned}
$$

From Eq. (A.6),

$$\begin{aligned}
\frac{\partial \mathcal{F}}{\partial Q(\boldsymbol{S}^{(m)'})} &= \sum_{\boldsymbol{S}^{(n)}} \sum_{\boldsymbol{d}^{(1)}} \sum_{\boldsymbol{d}^{(2)}} Q(\boldsymbol{S}^{(n)}) Q(\boldsymbol{d}^{(1)}) Q(\boldsymbol{d}^{(2)}) \ln P(\boldsymbol{O}|\boldsymbol{S}^{(m)'}, \boldsymbol{S}^{(n)}, \boldsymbol{d}^{(1)}, \boldsymbol{d}^{(2)}, \Lambda) \\
&\quad + \ln P(\boldsymbol{S}^{(m)'}|\Lambda_{S,m}) - \ln Q(\boldsymbol{S}^{(m)'}) - 1, \qquad \text{(A.11)} \\
\frac{\partial \mathcal{F}}{\partial Q(\boldsymbol{d}^{(m)'})} &= \sum_{\boldsymbol{S}^{(1)}} \sum_{\boldsymbol{S}^{(2)}} \sum_{\boldsymbol{d}^{(n)}} Q(\boldsymbol{S}^{(1)}) Q(\boldsymbol{S}^{(2)}) Q(\boldsymbol{d}^{(n)}) \ln P(\boldsymbol{O}|\boldsymbol{S}^{(1)}, \boldsymbol{S}^{(2)}, \boldsymbol{d}^{(m)'}, \boldsymbol{d}^{(n)}, \Lambda) \\
&\quad + \ln P(\boldsymbol{d}^{(m)'}|\Lambda_{S,m}) - \ln Q(\boldsymbol{d}^{(m)'}) - 1. \qquad \text{(A.12)}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\ln Q(\boldsymbol{S}^{(m)'}) &= \left\langle \ln P(\boldsymbol{O}|\boldsymbol{S}^{(m)'}, \boldsymbol{S}^{(n)}, \boldsymbol{d}^{(1)}, \boldsymbol{d}^{(2)}, \Lambda) \right\rangle_{Q(\boldsymbol{S}^{(n)})Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)})} + \ln P(\boldsymbol{S}^{(m)'}) \\
&\quad + const \\
\Leftrightarrow Q(\boldsymbol{S}^{(m)}) &\propto P(\boldsymbol{S}^{(m)}) \exp \left\langle \ln P(\boldsymbol{O}|\boldsymbol{S}^{(m)}, \boldsymbol{S}^{(n)}, \boldsymbol{d}^{(1)}, \boldsymbol{d}^{(2)}, \Lambda) \right\rangle_{Q(\boldsymbol{S}^{(n)})Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)})} \qquad \text{(A.13)} \\
\ln Q(\boldsymbol{d}^{(m)'}) &= \left\langle \ln P(\boldsymbol{O}|\boldsymbol{S}^{(1)}, \boldsymbol{S}^{(2)}, \boldsymbol{d}^{(n)}, \Lambda) \right\rangle_{Q(\boldsymbol{S}^{(1)})Q(\boldsymbol{S}^{(2)})(Q(\boldsymbol{d}^{(n)})} + \ln P(\boldsymbol{d}^{(m)'}) \\
&\quad + const \\
\Leftrightarrow Q(\boldsymbol{d}^{(m)}) &\propto P(\boldsymbol{d}^{(m)}) \exp \left\langle \ln P(\boldsymbol{O}|\boldsymbol{S}^{(m)}, \boldsymbol{S}^{(n)}, \boldsymbol{d}^{(1)}, \boldsymbol{d}^{(2)}, \Lambda) \right\rangle_{Q(\boldsymbol{S}^{(1)})Q(\boldsymbol{S}^{(2)})(Q(\boldsymbol{d}^{(n)})} \qquad \text{(A.14)}
\end{aligned}$$

where proportional symbol '∝' can be placed into equality sign '=' by introducing a normalizing constant for each of them.

A.1.1 Detail of approximated posterior distributions

In the following, the detail of $Q(\boldsymbol{S}^{(m)})$ is described. From the linearity of the expectation, it can be written as

$$\begin{aligned}
&\left\langle \sum_{t^{(1)}=1}^{T^{(1)}} \sum_{t^{(2)}=1}^{T^{(2)}} \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+d^{(1)}_{t^{(2)}}}, S^{(2)}_{t^{(2)}+d^{(2)}_{t^{(1)}}}, \Lambda) \right\rangle_{Q(\boldsymbol{S}^{(2)})Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)})} \\
&= \sum_{t^{(1)}=1}^{T^{(1)}} \sum_{t^{(2)}=1}^{T^{(2)}} \left\langle \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+d^{(1)}_{t^{(2)}}}, S^{(2)}_{t^{(2)}+d^{(2)}_{t^{(1)}}}, \Lambda) \right\rangle_{Q(\boldsymbol{S}^{(2)})Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)})}. \qquad \text{(A.15)}
\end{aligned}$$

New variables $\gamma^{(m)}$ and $\eta^{(m)}$ are introduced and can be defined as follows:

$$\begin{aligned}
\gamma^{(m)}(t^{(m)}, i) &= \sum_{\boldsymbol{S}^{(m)}} Q(\boldsymbol{S}^{(m)}) \delta(S^{(m)}_{t^{(m)}}, i) \qquad \text{(A.16)} \\
\eta^{(m)}(t^{(n)}, i) &= \sum_{\boldsymbol{d}^{(m)}} Q(\boldsymbol{d}^{(m)}) \delta(d^{(1)}_{t^{(n)}}, i), \quad n \neq m \qquad \text{(A.17)}
\end{aligned}$$

Using the above variables, the expectation in Eq. (A.15) can be written as follows:

$$
\begin{aligned}
&\left\langle \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+d^{(1)}_{t^{(2)}}}, S^{(2)}_{t^{(2)}+d^{(2)}_{t^{(1)}}}, \Lambda)\right\rangle_{Q(\boldsymbol{S}^{(2)})Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)})} \\
= & \sum_{\boldsymbol{S}^{(2)}}\sum_{\boldsymbol{d}^{(1)}}\sum_{\boldsymbol{d}^{(2)}} Q(\boldsymbol{S}^{(2)})Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)}) \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+d^{(1)}_{t^{(2)}}}, S^{(2)}_{t^{(2)}+d^{(2)}_{t^{(1)}}}, \Lambda) \\
= & \sum_{\boldsymbol{S}^{(2)}}\sum_{\boldsymbol{d}^{(1)}}\sum_{\boldsymbol{d}^{(2)}}\sum_{j=1}^{K^{(2)}} Q(\boldsymbol{S}^{(2)})Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)})\delta(S^{(2)}_{t^{(2)}+d^{(2)}_{t^{(1)}}}, j) \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+d^{(1)}_{t^{(2)}}}, j, \Lambda) \\
= & \sum_{j=1}^{K^{(2)}}\sum_{\boldsymbol{d}^{(1)}}\sum_{\boldsymbol{d}^{(2)}} Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)})\gamma^{(2)}(t^{(2)}+d^{(2)}_{t^{(1)}}, j) \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+d^{(1)}_{t^{(2)}}}, j, \Lambda) \qquad \text{(A.18)}
\end{aligned}
$$

Next, Eq. (A.18) can be re-written as follows.

$$
\begin{aligned}
&\sum_{k=1}^{K_d^{(1)}}\sum_{l=1}^{K_d^{(2)}}\sum_{j=1}^{K^{(2)}}\sum_{\boldsymbol{d}^{(1)}}\sum_{\boldsymbol{d}^{(2)}} Q(\boldsymbol{d}^{(1)})Q(\boldsymbol{d}^{(2)})\delta(d^{(1)}_{t^{(2)}}, k)\delta(d^{(2)}_{t^{(1)}}, l)\gamma^{(2)}(t^{(2)}+l, j) \\
&\quad \cdot \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+k}, j, \Lambda) \\
&= \sum_{k=1}^{K_d^{(1)}}\sum_{l=1}^{K_d^{(2)}}\sum_{j=1}^{K^{(2)}} \eta^{(1)}(t^{(2)}, k)\eta^{(2)}(t^{(1)}, l)\gamma^{(2)}(t^{(2)}+l, j) \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+k}, j, \Lambda)
\end{aligned}
$$

To simplify notation, following variable g is introduced.

$$
g(t^{(1)}, t^{(2)}, k, l, i, j) = \eta^{(1)}(t^{(2)}, k)\eta^{(2)}(t^{(1)}, l)\gamma^{(2)}(t^{(2)}+l, j) \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|i, j, \Lambda).
$$

As a result, Eq. (A.15) can be written temporarily as

$$
\begin{aligned}
&\sum_{t^{(1)}=1}^{T^{(1)}}\sum_{t^{(2)}=1}^{T^{(2)}}\sum_{k=1}^{K_d^{(1)}}\sum_{l=1}^{K_d^{(2)}}\sum_{j=1}^{K^{(2)}} \eta^{(1)}(t^{(2)}, k)\eta^{(2)}(t^{(1)}, l)\gamma^{(2)}(t^{(2)}+l, j) \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|S^{(1)}_{t^{(1)}+k}, j, \Lambda) \\
= & \sum_{\overline{t^{(1)}}}\sum_{\overline{t^{(2)}}}\sum_{t^{(1)}=1}^{T^{(1)}}\sum_{t^{(2)}=1}^{T^{(2)}}\sum_{k=1}^{K_d^{(1)}}\sum_{l=1}^{K_d^{(2)}}\sum_{j=1}^{K^{(2)}} \delta(\overline{t^{(1)}}, t^{(1)}+k)\delta(\overline{t^{(2)}}, t^{(2)}+l) g(t^{(1)}, t^{(2)}, k, l, i, j).
\end{aligned}
$$

(A.19)

Next, a part of summation in Eq. (A.19) can be written as

$$\begin{aligned}
&\sum_{k=1}^{K_d^{(1)}}\sum_{l=1}^{K_d^{(2)}}\sum_{j=1}^{K^{(2)}}\delta(\overline{t^{(1)}}, t^{(1)}+k)\delta(\overline{t^{(2)}}, t^{(1)}+l)g(t^{(1)}, t^{(2)}, k, l, i, j)\\
= \quad &\sum_{k=1}^{K_d^{(1)}}\sum_{l=1}^{K_d^{(2)}}\sum_{j=1}^{K^{(2)}}\delta(\overline{t^{(1)}}, t^{(1)}+k)\delta(\overline{t^{(2)}}, t^{(1)}+l)\eta^{(1)}(t^{(2)}, k)\eta^{(2)}(t^{(1)}, l)\gamma^{(2)}(\overline{t^{(2)}}, j)\\
&\cdot \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|i, j, \Lambda)\\
= \quad &\sum_{j=1}^{K^{(2)}}\eta^{(1)}(t^{(2)}, \overline{t^{(1)}}-t^{(1)})\eta^{(2)}(t^{(1)}, \overline{t^{(2)}}-t^{(2)})\gamma^{(2)}(\overline{t^{(2)}}, j)\ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|i, j, \Lambda)
\end{aligned}$$

Finally, Eq. (A.19) can be obtained as

$$\begin{aligned}
&\sum_{\overline{t^{(1)}}}\sum_{\overline{t^{(2)}}}\sum_{t^{(1)}=1}^{T^{(1)}}\sum_{t^{(2)}=1}^{T^{(2)}}\sum_{k=1}^{K_d^{(1)}}\sum_{l=1}^{K_d^{(2)}}\sum_{j=1}^{K^{(2)}}\delta(\overline{t^{(1)}}, t^{(1)}+k)\delta(\overline{t^{(2)}}, t^{(1)}+l)g(t^{(1)}, t^{(2)}, k, l, i, j)\\
= \quad &\sum_{\overline{t^{(1)}}}\sum_{\overline{t^{(2)}}}\sum_{t^{(1)}=1}^{T^{(1)}}\sum_{t^{(2)}=1}^{T^{(2)}}\sum_{j=1}^{K^{(2)}}\eta^{(1)}(t^{(2)}, \overline{t^{(1)}}-t^{(1)})\eta^{(2)}(t^{(1)}, \overline{t^{(2)}}-t^{(2)})\gamma^{(2)}(\overline{t^{(2)}}, j) \qquad \text{(A.20)}\\
&\cdot \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|i, j, \Lambda)\\
= \quad &\sum_{\overline{t^{(1)}}}\ln h_S^{(1)}(\overline{t^{(1)}}, S_{\overline{t^{(1)}}}^{(1)} = i), \qquad \text{(A.21)}
\end{aligned}$$

where

$$\begin{aligned}
\ln h_S^{(1)}(\overline{t^{(1)}}, i) \quad = \quad &\sum_{\overline{t^{(2)}}}\sum_{t^{(1)}=1}^{T^{(1)}}\sum_{t^{(2)}=1}^{T^{(2)}}\sum_{j=1}^{K^{(2)}}\eta^{(1)}(t^{(2)}, \overline{t^{(1)}}-t^{(1)})\eta^{(2)}(t^{(1)}, \overline{t^{(2)}}-t^{(2)})\gamma^{(2)}(\overline{t^{(2)}}, j)\\
&\cdot \ln P(\boldsymbol{O}_{t^{(1)}t^{(2)}}|i, j, \Lambda). \qquad \text{(A.22)}
\end{aligned}$$

Using the above $\ln h_S^{(m)}(\cdot, \cdot)$, $Q(\boldsymbol{S}^{(m)})$ can be obtained as

$$Q(\boldsymbol{S}^{(m)}) = \frac{1}{Z_S^{(m)}}P(\boldsymbol{S}^{(m)}|\Lambda^{(m)})\prod_{\overline{t^{(m)}}}h_S^{(m)}(\overline{t^{(m)}}, S_{\overline{t^{(m)}}}^{(m)} = i), \qquad \text{(A.23)}$$

where $Z_S^{(m)}$ is normalizing constant. $Q(\boldsymbol{d}^{(m)})$ can be written in a similar form as $Q(\boldsymbol{S}^{(m)})$.

A.2 Derivation of re-estimation formulas

The lower bound $\mathcal{F}$ in Eq. (A.5) can be factorized as

$$\begin{aligned}
\mathcal{F}(Q, \Lambda^{new}) &= \sum_{\boldsymbol{S}} \sum_{\boldsymbol{d}} Q(\boldsymbol{S})Q(\boldsymbol{d}) \ln P(\boldsymbol{O}|\boldsymbol{S}, \boldsymbol{d}, \Lambda^{new}) + \sum_{\boldsymbol{S}} Q(\boldsymbol{S}) \ln P(\boldsymbol{S}|\Lambda^{new}) \\
&\quad + \sum_{\boldsymbol{d}} Q(\boldsymbol{d}) \ln P(\boldsymbol{d}|\Lambda^{new}) - \sum_{\boldsymbol{S}} \sum_{\boldsymbol{d}} Q(\boldsymbol{S})Q(\boldsymbol{d}) \ln Q(\boldsymbol{S})Q(\boldsymbol{d}) \\
&= \mathcal{F}_b(Q, \Lambda^{new}) + \sum_{m=1,2} (\mathcal{F}_{S,\pi^{(m)}}(Q, \Lambda^{new}) + \mathcal{F}_{d,\pi^{(m)}}(Q, \Lambda^{new}) \\
&\quad + \mathcal{F}_{S,a^{(m)}}(Q, \Lambda^{new}) + \mathcal{F}_{d,a^{(m)}}(Q, \Lambda^{new})) - \sum_{\boldsymbol{S}} \sum_{\boldsymbol{d}} Q(\boldsymbol{S})Q(\boldsymbol{d}) \ln Q(\boldsymbol{S})Q(\boldsymbol{d}),
\end{aligned}$$

where

$$\begin{aligned}
\mathcal{F}_{S,\pi^{(m)}}(Q, \Lambda^{new}) &= \sum_{\boldsymbol{S}} Q(\boldsymbol{S}) \sum_{i=1}^{K^{(m)}} \delta(S_1^{(m)}, i) \ln \pi_{S,i}^{(m)} && \text{(A.24)} \\
&= \sum_{i=1}^{K^{(m)}} \left\langle S_1^{(m)}, i \right\rangle \cdot \ln \pi_{S,i}^{(m)}, && \text{(A.25)} \\
\mathcal{F}_{d,\pi^{(m)}}(Q, \Lambda^{new}) &= \sum_{\boldsymbol{d}} Q(\boldsymbol{d}) \sum_{i=1}^{K_d^{(m)}} \delta(d_1^{(m)}, i) \ln \pi_{d,i}^{(m)} && \text{(A.26)} \\
&= \sum_{i=1}^{K_d^{(m)}} \left\langle d_1^{(m)}, i \right\rangle \cdot \ln \pi_{d,i}^{(m)}, && \text{(A.27)}
\end{aligned}$$

$$\begin{aligned}
\mathcal{F}_{S,a^{(m)}}(Q, \Lambda^{new}) &= \sum_{\boldsymbol{S}} Q(\boldsymbol{S}) \sum_{t^{(m)}=2}^{T^{(m)}} \sum_{i=1}^{K^{(m)}} \sum_{j=1}^{K^{(m)}} \delta(S_{t^{(m)}-1}^{(m)}, i)\delta(S_{t^{(m)}}^{(m)}, j) \ln a_{S,ij}^{(m)} && \text{(A.28)} \\
&= \sum_{t^{(m)}=2}^{T^{(m)}} \sum_{i=1}^{K^{(m)}} \sum_{j=1}^{K^{(m)}} \left\langle (S_{t^{(m)}-1}^{(m)}, i)(S_{t^{(m)}}^{(m)}, j) \right\rangle \cdot \ln a_{S,ij}^{(m)}, && \text{(A.29)} \\
\mathcal{F}_{d,a^{(m)}}(Q, \Lambda^{new}) &= \sum_{\boldsymbol{d}} Q(\boldsymbol{d}) \sum_{t^{(n)}=2}^{T^{(n)}} \sum_{i=1}^{K_d^{(m)}} \sum_{j=1}^{K_d^{(m)}} \delta(d_{t^{(n)}-1}^{(m)}, i)\delta(d_{t^{(n)}}^{(m)}, j) \ln a_{d,ij}^{(m)} && \text{(A.30)} \\
&= \sum_{t^{(n)}=2}^{T^{(n)}} \sum_{i=1}^{K_d^{(m)}} \sum_{j=1}^{K_d^{(m)}} \left\langle (d_{t^{(m)}-1}^{(m)}, i)(d_{t^{(m)}}^{(m)}, j) \right\rangle \ln a_{d,ij}^{(m)}, && \text{(A.31)}
\end{aligned}$$

$$
\begin{aligned}
F_b(Q, \Lambda^{new}) &= -\frac{1}{2} \sum_{\boldsymbol{S},\boldsymbol{d}} Q(\boldsymbol{S}, \boldsymbol{d}) \sum_{t^{(1)}=1}^{T^{(1)}} \sum_{t^{(2)}=1}^{T^{(2)}} \sum_{i=1}^{K^{(1)}} \sum_{j=1}^{K^{(2)}} \sum_{k=1}^{K_d^{(1)}} \sum_{l=1}^{K_d^{(2)}} \delta(S_{t^{(1)}+d^{(1)}_{t^{(2)}}}, i) \\
&\quad \cdot \delta(S_{t^{(2)}+d^{(2)}_{t^{(1)}}}, j) \delta(d^{(1)}_{t^{(2)}}, k) \delta(d^{(2)}_{t^{(1)}}, l) \Big[\ln(2\pi)^D + \ln|\boldsymbol{\Sigma}_{ij}| \\
&\quad + (\boldsymbol{O}_{t^{(1)}t^{(2)}} - \boldsymbol{\mu}_{ij})' \boldsymbol{\Sigma}_{ij}^{-1} (\boldsymbol{O}_{t^{(1)}t^{(2)}} - \boldsymbol{\mu}_{ij}) \Big] \\
&= -\frac{1}{2} \sum_{t^{(1)}=1}^{T^{(1)}} \sum_{t^{(2)}=1}^{T^{(2)}} \sum_{i=1}^{K^{(1)}} \sum_{j=1}^{K^{(2)}} \sum_{k=1}^{K_d^{(1)}} \sum_{l=1}^{K_d^{(2)}} \left\langle S^{(1)}_{t^{(1)}+k}, i \right\rangle \left\langle S^{(2)}_{t^{(2)}+l}, j \right\rangle \left\langle d^{(1)}_{t^{(2)}}, k \right\rangle \\
&\quad \left\langle d^{(1)}_{t^{(2)}}, l \right\rangle \Big[\ln(2\pi)^D + \ln|\boldsymbol{\Sigma}_{ij}| + (\boldsymbol{O}_{t^{(1)}t^{(2)}} - \boldsymbol{\mu}_{ij})' \boldsymbol{\Sigma}_{ij}^{-1} (\boldsymbol{O}_{t^{(1)}t^{(2)}} - \boldsymbol{\mu}_{ij}) \Big].
\end{aligned}
$$

The re-estimation formulas of the transition probability can be obtained by maximizing the corresponding $\mathcal{F}$ based on the method of Lagrange multiplier. The re-estimation formulas of $\boldsymbol{\mu}_{ij}$ and $\boldsymbol{\Sigma}_{ij}$ can be obtained by taking partial derivatives of $\mathcal{F}_b$ for each parameters and put them equal to 0.

Appendix B

Derivation separable lattice trajectory 2-D HMMs

By imposing the explicit relationships between static and dynamic features represented by Eq. (5.15), Eq. (5.22) can be re-normalized and written as

$$\begin{aligned}
\mathcal{N}\left(\boldsymbol{W}\boldsymbol{C} \mid \boldsymbol{\mu}_S, \boldsymbol{\Sigma}_S\right) &= \frac{1}{\sqrt{(2\pi)^{3MT}\left|\boldsymbol{\Sigma}_S\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{W}\boldsymbol{C}-\boldsymbol{\mu}_S\right)^{\top}\boldsymbol{\Sigma}_S\left(\boldsymbol{W}\boldsymbol{C}-\boldsymbol{\mu}_S\right)\right\} && \text{(B.32)}\\
&= \frac{1}{\sqrt{(2\pi)^{3MT}\left|\boldsymbol{\Sigma}_S\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_S^{\top}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{\mu}_S + \boldsymbol{C}^{\top}\boldsymbol{W}^{\top}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{W}\boldsymbol{C} - 2\boldsymbol{\mu}_S^{\top}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{W}\boldsymbol{C}\right)\right\} && \text{(B.33)}\\
&= \frac{1}{\sqrt{(2\pi)^{3MT}\left|\boldsymbol{\Sigma}_S\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_S^{\top}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{\mu}_S + \boldsymbol{C}^{\top}\boldsymbol{R}_S\boldsymbol{C} - 2\boldsymbol{r}_S^{\top}\boldsymbol{C}\right)\right\} && \text{(B.34)}\\
&= \frac{1}{\sqrt{(2\pi)^{3MT}\left|\boldsymbol{\Sigma}_S\right|}} \exp\left[-\frac{1}{2}\left\{\left(\boldsymbol{C}-\overline{\boldsymbol{C}}_S\right)^{\top}\boldsymbol{R}_S\left(\boldsymbol{C}-\overline{\boldsymbol{C}}_S\right) - \boldsymbol{r}_S^{\top}\boldsymbol{P}_S\boldsymbol{r}_S + \boldsymbol{\mu}_S^{\top}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{\mu}_S\right\}\right] && \text{(B.35)}\\
&= \frac{\sqrt{(2\pi)^{MT}\left|\boldsymbol{P}_S\right|}}{\sqrt{(2\pi)^{3MT}\left|\boldsymbol{\Sigma}_S\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{\mu}_S^{\top}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{\mu}_S - \boldsymbol{r}_S^{\top}\boldsymbol{P}_S\boldsymbol{r}_S\right)\right\}\\
&\qquad \times \frac{1}{\sqrt{(2\pi)^{MT}\left|\boldsymbol{P}_S\right|}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{C}-\overline{\boldsymbol{C}}_S\right)^{\top}\boldsymbol{R}_S\left(\boldsymbol{C}-\overline{\boldsymbol{C}}_S\right)\right\} && \text{(B.36)}\\
&= Z_S \cdot \mathcal{N}\left(\boldsymbol{C} \mid \overline{\boldsymbol{C}}_S, \boldsymbol{P}_S\right), && \text{(B.37)}
\end{aligned}$$

where $T = T^{(1)}T^{(2)}$, $\boldsymbol{R}_S = \boldsymbol{W}^{\top}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{W} = \boldsymbol{P}_S^{-1}$, $\boldsymbol{r}_S = \boldsymbol{W}^{\top}\boldsymbol{\Sigma}_S^{-1}\boldsymbol{\mu}_S$, and $\overline{\boldsymbol{C}}_S = \boldsymbol{P}_S\boldsymbol{r}_S$.

Based on the above relationship and Eq. (5.31), SL2D-HMMs can be re-defined as follows:

$$
\begin{aligned}
P(\boldsymbol{C} \mid \Lambda) &= \sum_{\boldsymbol{S}} P(\boldsymbol{C}, \boldsymbol{S} \mid \Lambda) = \sum_{\boldsymbol{S}} P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) P(\boldsymbol{S} \mid \Lambda) && \text{(B.38)} \\
&= \sum_{\boldsymbol{S}} P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) \prod_{m=1,2} P(\boldsymbol{S}^{(m)} \mid \Lambda), && \text{(B.39)}
\end{aligned}
$$

where $P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) = \mathcal{N}\left(\boldsymbol{C} \mid \overline{\boldsymbol{C}}_{\boldsymbol{S}}, \boldsymbol{P}_{\boldsymbol{S}}\right)$.

Appendix C

Derivation of reestimation formula for the concatenated mean vector $\boldsymbol{m}$

From Eq. (5.49), the log-likelihood (Eq. (5.43)) can be written as follows:

$$\log P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda) \approx -\frac{1}{2}\left\{\boldsymbol{m}^\top \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}^\top \boldsymbol{P}_S \boldsymbol{W} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{F}_S \boldsymbol{m} - 2\boldsymbol{m}^\top \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}^\top \boldsymbol{C}\right\}, \quad \text{(C.40)}$$

where the terms not containing $\boldsymbol{m}$ are omitted. By taking a partial derivative of $\boldsymbol{m}$, the gradient function of $\boldsymbol{m}$ can be obtained as

$$\begin{aligned}
\frac{\partial \log P(\boldsymbol{C} \mid \boldsymbol{S}, \Lambda)}{\partial \boldsymbol{m}} &= -\boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}^\top \boldsymbol{P}_S \boldsymbol{W} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{F}_S \boldsymbol{m} + \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}^\top \boldsymbol{C} && \text{(C.41)} \\
&= \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}\left(\boldsymbol{C} - \boldsymbol{P}_S \boldsymbol{W} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{F}_S \boldsymbol{m}\right) && \text{(C.42)} \\
&= \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}\left(\boldsymbol{C} - \boldsymbol{P}_S \boldsymbol{W} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{\mu}_S\right) && \text{(C.43)} \\
&= \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}\left(\boldsymbol{C} - \boldsymbol{P}_S \boldsymbol{r}_S\right) && \text{(C.44)} \\
&= \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}\left(\boldsymbol{C} - \overline{\boldsymbol{C}}_S\right) && \text{(C.45)}
\end{aligned}$$

By setting Eq. (C.41) equal to $\boldsymbol{0}_{3MK^{(1)}K^{(2)}}$, the re-estimation formula of $\boldsymbol{m}$ can be obtained as follows:

$$\begin{aligned}
-\boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}^\top \boldsymbol{P}_S \boldsymbol{W} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{F}_S \boldsymbol{m} + \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}^\top \boldsymbol{C} &= \boldsymbol{0}_{3MK^{(1)}K^{(2)}} && \text{(C.46)} \\
\boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}^\top \boldsymbol{P}_S \boldsymbol{W} \boldsymbol{\Sigma}_S^{-1} \boldsymbol{F}_S \boldsymbol{m} &= \boldsymbol{F}_S^\top \boldsymbol{\Sigma}_S^{-1} \boldsymbol{W}^\top \boldsymbol{C} && \text{(C.47)} \\
\therefore \hat{\boldsymbol{m}} &= \boldsymbol{A}^{-1} \boldsymbol{b}, && \text{(C.48)}
\end{aligned}$$

where ***A*** and ***b*** are defined as

$$\boldsymbol{A} = \boldsymbol{G}_{\boldsymbol{S}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{W}\boldsymbol{P}_{\boldsymbol{S}}\boldsymbol{W}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{G}_{\boldsymbol{S}}, \tag{C.49}$$

$$\boldsymbol{b} = \boldsymbol{G}_{\boldsymbol{S}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{S}}^{-1}\boldsymbol{W}\boldsymbol{C}. \tag{C.50}$$