

BOSTON UNIVERSITY  
GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

**EXPLOITING PHONOLOGICAL CONSTRAINTS FOR HANDSHAPE  
RECOGNITION IN SIGN LANGUAGE VIDEO**

by

**ASHWIN THANGALI VARADARAJU**

B.E., National Institute of Technology, Surathkal, India  
M.E., Indian Institute of Science, Bangalore, India

Submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy

2013

© Copyright by  
ASHWIN THANGALI VARADARAJU  
2013



Approved by

First Reader

---

Stan Sclaroff, PhD  
Professor of Computer Science  
Boston University

Second Reader

---

Carol Neidle, PhD  
Professor of French and Linguistics  
Boston University

Third Reader

---

Margrit Betke, PhD  
Professor of Computer Science  
Boston University

Fourth Reader

---

Erik B. Sudderth, PhD  
Assistant Professor of Computer Science  
Brown University

# EXPLOITING PHONOLOGICAL CONSTRAINTS FOR HANDSHAPE RECOGNITION IN SIGN LANGUAGE VIDEO

(Order No.                    )

**ASHWIN THANGALI VARADARAJU**

Boston University, Graduate School of Arts and Sciences, 2013

Major Professor: Stan Sclaroff, Department of Computer Science

## ABSTRACT

The ability to recognize handshapes in sign language video is essential in algorithms for sign recognition and retrieval. Handshape recognition from isolated images is, however, an insufficiently constrained problem. Many handshapes share similar 3D configurations and are indistinguishable for some hand orientations in 2D image projections. Additionally, significant differences in handshape appearance are induced by the articulated structure of the hand and variants produced by different signers. Linguistic rules involved in the production of signs impose strong constraints on the articulations of the hands, yet little attention has been paid to exploiting these constraints in previous works on sign recognition.

The focus of this research is American Sign Language (ASL), although the same approach could be applied to other signed languages. Among the different classes of signs in ASL, so-called “lexical signs” constitute the prevalent class. Morphemes (i.e., meaningful units) for signs in this class involve a combination of particular handshapes, palm orientations, locations for articulation, and movement type. These are analyzed by many sign linguists as analogues of phonemes in spoken languages. As in spoken language, phonological constraints govern the ways in which phonemes combine in signed languages; utilizing these constraints for handshape recognition in ASL is the focus of this thesis.

Handshapes in monomorphemic lexical signs are specified at the start and end of the sign. Handshape transitions within a sign are generally constrained to involve either closing

or opening of the hand (i.e., folding or unfolding of the palm and one or more fingers). Akin to allophonic variations in spoken languages, both inter- and intra- signer variations in the production of specific handshapes are observed. We propose a Bayesian network formulation to exploit handshape co-occurrence constraints, also utilizing information about allophonic variations to aid in handshape recognition. We propose a fast non-rigid image alignment method to gain improved robustness to handshape appearance variations during computation of observation likelihoods in the Bayesian network.

We evaluate our handshape recognition approach on a large dataset of monomorphemic lexical signs. We demonstrate that leveraging linguistic constraints on handshapes results in improved handshape recognition accuracy. As part of the overall project, a large corpus is being prepared for dissemination: video for three thousand signs, each from up to six native signers of ASL, annotated with linguistic information such as glosses and morpho-phonological properties and variations, including the start/end handshapes associated with each ASL sign production.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Aims of this dissertation . . . . .	7
1.2	Overview of the proposed formulation for handshape inference . . . . .	9
1.3	The lexicon video dataset for ASL . . . . .	10
1.4	Summary of contributions . . . . .	11
1.5	Summary of results . . . . .	13
1.6	Thesis roadmap . . . . .	13
1.7	Publications reporting research conducted in this thesis . . . . .	14
<b>2</b>	<b>Background about American Sign Language (ASL)</b>	<b>15</b>
2.1	Introduction to sign language . . . . .	15
2.1.1	Challenges that arise in developing linguistic models of sign language	15
2.2	Linguistic organization of signed languages . . . . .	16
2.2.1	Units of meaning and articulatory units conveying linguistic distinctions in (signed / spoken) language . . . . .	16
2.2.2	The internal structures of morphemes . . . . .	17
2.2.3	Morphological subclasses of signs in ASL . . . . .	17
2.3	Handshapes in monomorphemic lexical signs . . . . .	20
2.3.1	Handshape representation . . . . .	20
2.3.2	Linguistic constraints governing handshape articulation in monomorphemic lexical signs . . . . .	24
2.3.3	Handshape variation in monomorphemic lexical signs . . . . .	27
2.4	Summary . . . . .	32

<b>3</b>	<b>Related Work</b>	<b>34</b>
3.1	HMM models for Sign Language Recognition . . . . .	36
3.2	Tracking hand articulations in general hand gestures . . . . .	40
3.3	Handshape recognition in sign language . . . . .	42
3.4	Appearance features for handshape verification . . . . .	44
3.5	Summary . . . . .	45
<b>4</b>	<b>ASL Lexicon Video Dataset (ASLLVD)</b>	<b>47</b>
4.1	Objectives and requirements for the lexicon dataset . . . . .	47
4.2	Data collection and annotation . . . . .	48
4.2.1	Native ASL signers provide signs for the dataset . . . . .	48
4.2.2	Video capture setup . . . . .	48
4.2.3	Elicitation methods . . . . .	49
4.2.4	Annotation methods . . . . .	49
4.3	The ASLLVD corpus . . . . .	51
4.3.1	Limitations in relying on handshape annotations as the ground-truth	59
4.4	Summary . . . . .	59
<b>5</b>	<b>HandShapes Bayesian Network (HSBN)</b>	<b>62</b>
5.1	HSBN for one-handed signs . . . . .	63
5.2	HSBN for two-handed signs . . . . .	65
5.3	Handshape inference using the HSBN model . . . . .	67
5.3.1	Handshape inference in one-handed signs . . . . .	67
5.3.2	Handshape inference in two-handed signs . . . . .	69
5.4	Summary . . . . .	70
<b>6</b>	<b>Learning the HSBN model</b>	<b>71</b>
6.1	The MAPEM formulation for learning HSBN <sup>dominant</sup> model parameters . . .	73
6.2	Variational Bayes formulation for learning HSBN <sup>dominant</sup> model parameters	82
6.3	Learning the HSBN <sup>congruent</sup> model parameters . . . . .	92

6.3.1	The MAPEM formulation for learning HSBN <sup>congruent</sup> model parameters	94
6.3.2	The VBEM formulation for learning HSBN <sup>congruent</sup> model parameters	95
6.4	Summary	97
<b>7</b>	<b>Learning a State-space for Hidden Variables in the HSBN</b>	<b>101</b>
7.1	Overview of the HSBNStateSpaceEstimation algorithm	103
7.2	Initializing the HSBN state-space	105
7.3	Hyper-parameters for the prior distributions	105
7.4	HSBN state-space refinement	106
7.5	Summary	110
<b>8</b>	<b>Handshape image observation likelihood model</b>	<b>119</b>
8.1	Computing the handshape observation likelihood	120
8.2	Computing the handshape appearance similarity score, $\text{sim}(\mathbf{i}, \mathbf{j})$	120
8.2.1	Background	121
8.2.2	Proposed formulation for non-rigid image alignment	122
8.2.3	Illustration of alignment results using the proposed algorithm	128
8.3	Summary	129
<b>9</b>	<b>Experiments: Implementation</b>	<b>131</b>
9.1	Training, retrieval and test sets for HSBN evaluation	131
9.2	Learning the HSBN	135
9.2.1	Hyper-parameters for prior distributions	135
9.2.2	State-space refinement using the HSBNStateSpaceSelection algorithm	145
9.3	Handshape retrieval to compute observation likelihoods	146
9.3.1	Pre-processing of hand images	146
9.3.2	Local feature representation	147
9.3.3	Computing the non-rigid image alignment	147
9.3.4	Filter and refine handshape retrieval	148
9.4	Handshape inference using the HSBN	148

9.5	Evaluating HSBN handshape inference performance . . . . .	149
<b>10</b>	<b>Experiments: Results</b>	<b>151</b>
10.1	Learning the HSBN . . . . .	151
10.2	Handshape retrieval using image alignment . . . . .	164
10.3	Handshape inference using the HSBN . . . . .	164
10.3.1	Performance summarized for all handshape classes . . . . .	166
10.3.2	Performance analyzed for each handshape class . . . . .	170
10.3.3	Performance analyzed for two different articulatory classes . . . . .	173
10.3.4	Performance analyzed through the learning epochs . . . . .	174
10.3.5	Examples illustrating HSBN handshape inference results . . . . .	176
10.4	Discussion . . . . .	180
<b>11</b>	<b>Discussion and Future work</b>	<b>183</b>
11.1	Limitations of the proposed formulation . . . . .	185
11.1.1	Assumptions in the HSBN representation . . . . .	185
11.1.2	Learning the HSBN model . . . . .	185
11.1.3	Observation likelihoods for start/end hand images . . . . .	187
11.2	Future work . . . . .	188
11.2.1	Enhancements to the formulation . . . . .	188
11.2.2	Empirical assessment . . . . .	191
11.3	Summary . . . . .	192
	<b>References</b>	<b>194</b>

## List of Tables

3.1	A review of approaches for handshape recognition (in sign language) and handpose tracking (general hand gestures). . . . .	46
4.1	Statistics for signs contained in the ASLLVD corpus. . . . .	52
4.2	Handshape labels and attested variants among examples of ACCIDENT and APPOINTMENT contained in the lexicon dataset. . . . .	55
4.3	A summary of different variations observed in the lexicon dataset for the signs ACCIDENT and APPOINTMENT based on annotations for different productions of these signs as listed in Table 4.2. . . . .	55
5.1	Notations used in the HSBN formulation. . . . .	63
5.2	Parameters for the HSBN formulation. . . . .	65
6.1	Notations for the training set and hidden variables employed in learning the HSBN <sup>dominant</sup> model. . . . .	72
6.2	Parameters in the HSBN learning formulation. . . . .	74
6.3	Summary of the MAPEM formulation for learning the HSBN. . . . .	99
6.4	Summary of the VBEM formulation for learning the HSBN. . . . .	99
9.1	Statistics for the productions of monomorphemic lexical signs from six native signers contained in the HSBN training, retrieval and test sets. . . . .	132
9.2	Statistics for the number of distinct monomorphemic lexical signs in the HSBN training, retrieval and test sets. . . . .	132
9.3	Statistics for the different articulatory classes contained in the HSBN training, retrieval and test sets. . . . .	132



9.4	Table of start/end handshape co-occurrences computed from handshapes on the dominant hand in one-handed and two-handed : different handshapes signs, and, from handshapes on the dominant and non-dominant hands in two-handed : same handshapes signs. . . . .	142
9.5	Start/end handshape labels from different productions of lexical items in the ASLLVD are used to construct the table of handshapes that are observed to have been produced in free variation with other handshapes. . . . .	143
9.6	Examples of cells from the handshape variants table (Table 9.5) that were annotated by the author as 'primary' and 'secondary' variants are displayed in the left and right tables respectively. . . . .	144
10.1	Nearest neighbor handshape retrieval results illustrated in the top plot are summarized in the above table. The highest recognition scores are highlighted in red. . . . .	165
10.2	Simple-NN retrieval and HSNB handshape inference results for the entire test set shown in the top plot are summarized in the above table. The highest recognition scores are highlighted in red. . . . .	167
10.3	Handshape retrieval/inference results for one-handed signs and two-handed : same handshapes signs displayed in the top plot are summarized in the above table. The highest recognition scores are highlighted in red. . . . .	172

## List of Figures

1.1	Computer vision applications for sign language recognition. . . . .	4
1.2	Examples illustrating challenges involved in identifying handshapes from hand images in ASL video. Signs in these video sequences were produced by a native signer. . . . .	5
1.3	An example illustrating the handshape inference problem in monomorphemic lexical signs solved by the HSBN model formulated in this thesis. The signer’s right hand has been chosen here for the handshape inference problem to simplify this illustration. . . . .	8
2.1	Examples for pairs of signs minimally distinguished by handshape, hand location, hand orientation, or, hand movement trajectory. . . . .	18
2.2	Examples of compound (or, polymorphemic) signs. . . . .	19
2.3	The 85 handshapes in ASL labeled according to annotation conventions in [Neidle, 2007]. . . . .	23
2.4	A taxonomy of constraints on handshapes for monomorphemic lexical signs in ASL [Battison, 2000]. The percentages (and total numbers of signs) in the ASLLVD collection corresponding to each constraint are also shown.	25
2.5	Changes in hand configuration within monomorphemic lexical signs are constrained to involve either closing or opening of the hand (i.e., the folding/unfolding of the palm and a selected subset of fingers). A sign from the class of <i>loan signs</i> that violates this constraint is shown here. . . . .	26
2.6	Examples of pairs of signs exhibiting sign-specific variations in handshape on the dominant or non-dominant hand. . . . .	29

2.7	Examples of phonological variation in handshape. Variations in this class are not specific to a particular lexical item. . . . .	31
4.1	Annotations in the ASLLVD delineate variations in articulatory features that are linguistically distinctive. A few examples of such distinctions in different articulatory parameters are shown here. . . . .	56
4.2	Lexical and phonological variations attested for the sign ACCIDENT in the ASLLVD. . . . .	57
4.3	Lexical and phonological variations attested for the sign APPOINTMENT in the ASLLVD. . . . .	58
4.4	Examples of handshape variation attested in the lexicon dataset that are exploited in learning the HSBN model. . . . .	60
5.1	The HSBN <sup>dominant</sup> graphical model for handshape inference in one-handed signs. . . . .	63
5.2	The HSBN <sup>congruent</sup> graphical model formulated for handshape inference in two-handed : same handshapes signs. . . . .	66
5.3	The HSBN <sup>dominant</sup> and HSBN <sup>non-dominant</sup> graphical models formulated for handshape inference in two-handed : different handshapes signs. . . . .	66
6.1	A plate representation for start/end handshape labels annotated for one-handed signs contained in the training set to learn the HSBN <sup>dominant</sup> model. . . . .	73
6.2	A plate representation for start/end handshape labels of the dominant and non-dominant hands annotated for two-handed : same handshapes signs contained in the training set to learn the HSBN <sup>congruent</sup> model. . . . .	73
7.1	An overview of the proposed optimization formulation for learning the HSBN parameters. . . . .	104

7·2	An illustration of results produced using the proposed algorithm for learning the HSBN. Parameters obtained after model initialization are displayed in the left column. The state-space refinement methods employed to generate model candidates in each epoch are listed in the center column. The estimated start/end latent states and model parameters in the final epoch after convergence of the variational Bayes lower bound are displayed in the last column. . . . .	110
8·1	Computing a bi-directional alignment for an example handshape image pair $(\mathbf{i}, \mathbf{j})$ . . . . .	128
9·1	An illustration of a few different choices that are possible in defining the hyper-parameters, $\beta_{\tau=1}^{\text{prior}}$ , of Dirichlet priors in the first learning epoch for the multinomial emission distribution parameters, $\mathbf{b}_{\tau=1}$ . . . . .	137
9·2	The hyper-parameters, $\alpha_{\tau=1}^{\text{prior}}, (\beta_{\tau=1}^{\text{s prior}}, \beta_{\tau=1}^{\text{e prior}})$ , specified for prior distributions associated with the model parameters $\mathbf{a}, \mathbf{b}^{\text{s}}, \mathbf{b}^{\text{e}}$ . . . . .	138
10·1	The normalized values for the hyper-parameters, $\nu_{\tau=1}^*, \alpha_{\tau=1}^*, \beta_{\tau=1}^{\text{s}*}, \beta_{\tau=1}^{\text{e}*}$ , estimated in the first learning epoch using the VBEM algorithm. . . . .	152
10·2	Estimated values for the VBEM lower bound produced by the sequence of state-space refinements in the HSBNStateSpaceEstimation algorithm. . . .	153
10·3	The normalized values for the hyper-parameters, $\nu_{\tau=180}^*, \alpha_{\tau=180}^*, \beta_{\tau=180}^{\text{s}*}, \beta_{\tau=180}^{\text{e}*}$ , that were estimated in the final learning epoch of the HSBNStateSpaceEstimation algorithm. . . . .	156
10·4	Normalized hyper-parameter values, $\beta_{\tau=180}^{\text{s}*}, \beta_{\tau=180}^{\text{e}*}$ , for emission distributions of start and end latent states estimated in the <i>final</i> epoch – part 1 of 3. . . . .	158
10·5	Normalized hyper-parameter values, $\beta_{\tau=180}^{\text{s}*}, \beta_{\tau=180}^{\text{e}*}$ , for emission distributions of start and end latent states estimated in the <i>final</i> epoch – part 2 of 3. . . . .	159

10·6	Normalized hyper-parameter values, $\beta_{\tau=180}^{s*}$ , $\beta_{\tau=180}^{e*}$ , for emission distributions of start and end latent states estimated in the <i>final</i> epoch – part 3 of 3. The start/end latent state indices for which a one-to-one association were not obtained are displayed in gray. . . . .	160
10·7	Normalized hyper-parameter values, $\alpha_{\tau=180}^*$ , for start $\rightarrow$ end latent state transitions estimated in the <i>final</i> epoch – part 1 of 3. . . . .	161
10·8	Normalized hyper-parameter values, $\alpha_{\tau=180}^*$ , for start $\rightarrow$ end latent state transitions estimated in the <i>final</i> epoch – part 2 of 3. . . . .	162
10·9	Normalized hyper-parameter values, $\alpha_{\tau=180}^*$ , for start $\rightarrow$ end latent state transitions estimated in the <i>final</i> epoch – part 3 of 3. The start/end latent state indices for which a one-to-one association were not obtained are displayed in gray. . . . .	163
10·10	Results of simple nearest neighbor handshape retrieval using different image alignment methods to compute the similarity scores. . . . .	165
10·11	Performance of HSBN handshape inference summarized over all handshape classes. . . . .	167
10·12	Impact of the value of $\beta$ in the observation likelihood model on handshape inference accuracy. . . . .	168
10·13	Evaluating handshape inference performance for each of the different handshape classes contained in the HSBN test set. . . . .	169
10·14	A listing of the handshape classes whose indices appear on the x-axis in the charts displayed in Figure 10·13. . . . .	170
10·15	Handshape inference performance for one-handed query signs are compared to the handshape inference performance for two-handed:same handshapes signs. . . . .	172
10·16	A comparison of frequencies for different handshapes observed in the one-handed and two-handed:same handshapes signs that are contained in the HSBN test and retrieval sets. . . . .	173

10·17	Evaluation of the test set handshape inference accuracy as a function of the learning epochs employed for state-space refinement in the HSBNStateSpaceEstimation algorithm. . . . .	175
10·18	Examples of results for start/end handshape inference in one-handed signs using the HSBN. . . . .	178
10·19	Examples of results for start/end handshape inference in two-handed:same handshapes signs using the HSBN <sup>congruent</sup> model. . . . .	179
11·1	An example of a query sign where HSBN handshape inference fails to produce acceptable results because nearest neighbor retrieval for each of the query hand images does not succeed in retrieving the correct handshape among the top-200 results. . . . .	187
11·2	An alternate formulation of the HSBN to more directly represent the dependencies between the handshapes articulated on the non-dominant hand and that of the dominant hand in two-handed:same handshapes signs. . .	190

## List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
AAM	Active Appearance Models
ASL	American Sign Language
ASLLVD	American Sign Language Lexicon Video Dataset
ASLLRP	American Sign Language Linguistics Research Project
DBN	Dynamic Bayesian Network
EM	Expectation-Maximization
HCI	Human Computer Interfaces
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HSBN	Hand Shapes Bayesian Network
LBP	Loopy Belief Propagation algorithm
LSE	Linear Systems of Equations
MAP	Maximum A-Posteriori Estimate
MRF	Markov Random Field
MRF-LBP	Markov Random Field formulation using the Loopy Belief Propagation algorithm

## List of Abbreviations, continued

NN	Nearest Neighbor retrieval
PCA	Principal Components Analysis
RANSAC	Random Sampling and Consensus algorithm
SIFT	Scale-Invariant Feature Transform
SLR	Sign Language Recognition
SVM	Support Vector Machines classifier
VB	Variational Bayes
VBEM	Variational Bayes Expectation Maximization Algorithm
MAPEM <sup>dominant</sup>	Maximum A-Posteriori (MAP) Expectation Maximization algorithm for training the HSBN <sup>dominant</sup> model
VBEM <sup>dominant</sup>	Variational Bayes formulation for training HSBN <sup>dominant</sup>
MAPEM <sup>congruent</sup>	MAPEM algorithm for training the HSBN <sup>congruent</sup> model
VBEM <sup>congruent</sup>	VBEM algorithm for training HSBN <sup>congruent</sup>



## Acknowledgments

My thesis advisors, Stan Sclaroff and Carol Neidle, have played an inalienable role in my development as a researcher. Their help, guidance and hands-on collaboration on the research project leading to this dissertation as well as their patient reading of this manuscript along with extensive suggestions for improving the presentation were crucial in my being able to make progress towards a defensible thesis. Completing this dissertation would have been inconceivable without their continuous encouragement and confidence in my ability to finish.

I would like to thank my thesis committee members for their feedback on the many early drafts of this dissertation and for their invaluable help in guiding the direction of this research towards fertile areas suitable for in-depth investigation. Erik Sudderth's suggestions motivated me to dig deeper into questions pertaining to the learning of latent variable models. Margrit Betke guided me all throughout my PhD study especially with suggestions to ensure that I focus on the most salient aspects in presenting my research. George Kollios was one of my first mentors in the PhD program and has always stood out for being a kind, genial and generous advisor.

The work reported here has been supported in part by grants from the National Science Foundation (IIS-0705749, CNS-0855065, IIS-0964385, CNS-1059218).

Among the many computer science professors with whom I have had fruitful discussions and who have gone out of their way to offer timely advice include Jonathan Appavoo, John Byers, Mark Crovella, Peter Gacs, Steve Homer, Abraham Matta, Evimaria Terzi, and, Rich West.

I have fond memories of P. S. Sastry and Sugata Ghosal who have mentored me in my previous research conducted at the Indian Institute of Science, Bangalore and at the IBM Research Lab, New Delhi.

Colleagues in the Image and Video Computing group who have graciously accepted to participate in my practice presentations, provided very substantial amounts of feedback, provided me an opportunity to collaborate on different research projects, inspired new

ideas, and, from whom I have learnt a whole lot during the past several years include Vitaly Ablavsky, Jonathon Alon, Vassilis Athitsos, Qinxun Bai, Mike Breslav, Fatih Cakir, Gokberk Cinbis, Sam Epstein, Murat Erdem, Wenxin Feng, Danna Gurari, Toni Hernandez, Nazli Ikizler-Cinbis, John Isidoro, Tianxiong Jiang, Ajjen Joshi, Rui Li, Liliana Lo Presti, He Kun, Shugao Ma, John Magee, Eric Missimer, Bill Mullaly, Walter Nunziati, Alexandra Stefan, Tian Taipeng, Diane Theriault, Jingbin Wang, Gary Wong, Zheng Wu, Quan Yuan, and, Jianming Zhang.

I have also had the rewarding experience of working closely with several students and researchers in the linguistics program. A partial list includes, Rachel Benedict, Alix Kraminitz, Jaimee DiMarco, Joan Nash, Indya Oliver, Caelen Pacelli, Braden Painter, Chrisann Papera, Donna Riggle, Tory Sampson, Jessica Scott, Jon Suen, Amelia Wisniewski-Barker, and, Iryna Zhuravlova.

Computer science department staff who have been incredibly helpful in taking care of all the niggles be it pertaining to computer systems or to administrative issues include, Nora Conroy, Christopher Devits, Ellen Grady, Wesley Harrell, Paul Stauffer, Jennifer Streubel, Theresa Sullivan, Joseph Szep, and, Austin Wolfe.

Thanks also to friends Priya Bangal, Vishesh Duggar, Usha Guduri, Krishna Kumar Subramanian, Praveen Pilly, Shreya Tiwari who have been steadfast in pushing me to finish my PhD all these years.

My wife, ♡Meghna Dilip♡, has had to essentially live through a second PhD and I thank her for her patience, love and support.

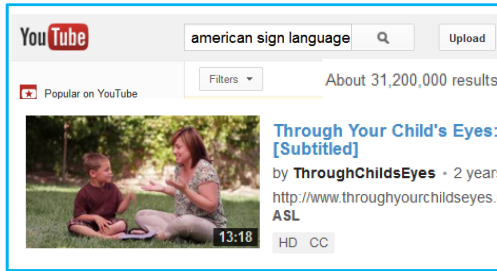
Finally, I would like to thank my parents, grandparents, brother as well as Meghna's parents for their unconditional love, support and encouragement. Without the support of my family I would most assuredly never have made it thus far.

## Chapter 1

### Introduction

Signed languages are visual-gestural languages used as the primary means of communication by the Deaf (individuals who culturally identify with the Deaf community). Signed languages are full-fledged natural languages, which are generally quite different from spoken languages that are used in the same region. Applications analogous to those enabled by speech recognition have been envisioned by computer vision researchers for signed languages, as well. The broad spectrum of possible applications includes sign language video retrieval and recognition given video input. Query-by-text is not often suitable for searching sign language video content because annotations in the form of subtitles and/or video transcriptions are only available in a small fraction of sign language video sequences. Spoken language subtitles, where available, do not exactly match the sign language source because of the substantial differences between the two languages. The ability to search sign language content using a query-by-sign interface can significantly improve access to sign language users for the sign language video collections that are available today. A sign language recognition (SLR) system in general needs to be able to detect, identify and recognize signs that are contained in the input signing video. Despite the importance of research in SLR and the substantial progress that has been demonstrated in advancing the state of the art in this area ([Cooper et al., 2011] presents a recent survey of computer vision approaches for SLR), person-independent recognition and retrieval of signs produced in natural environments remains a challenging problem, particularly when a large vocabulary of signs are involved.

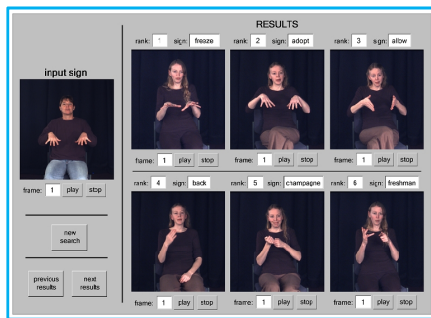
One application of specific interest in our research is a query-by-sign search interface for a sign language dictionary. In the envisioned system the user can search the dictionary for



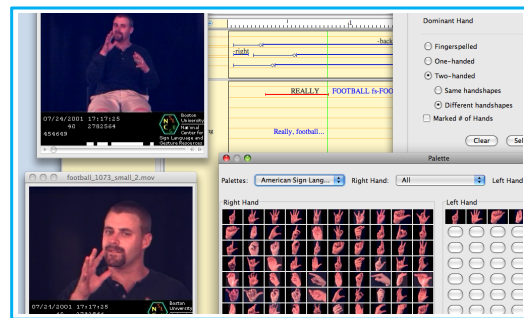
Sign language video retrieval



Recognizing signs contained in a continuous signing video sequence



Query-by-sign sign language dictionary



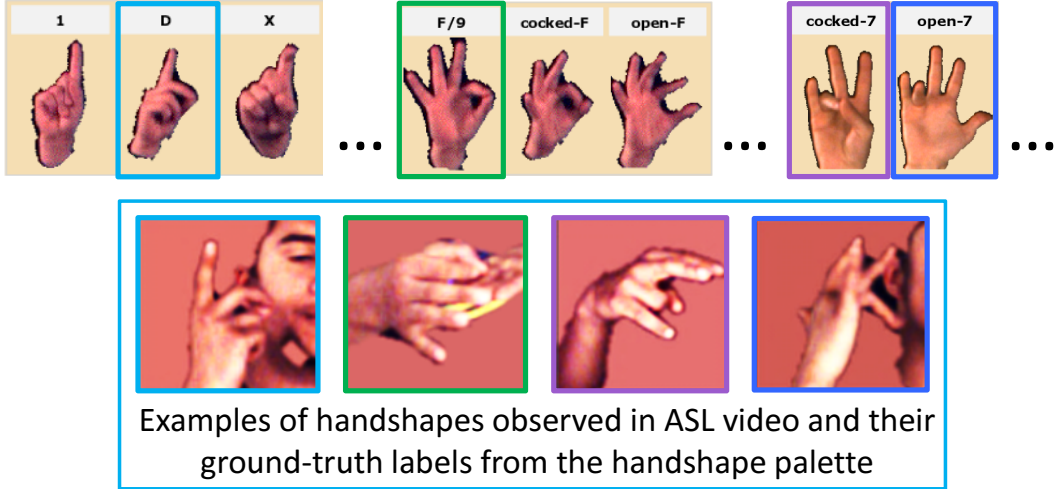
Assist in preparing linguistic annotations

**Figure 1.1:** Computer vision applications for sign language recognition.

a specific sign, as produced in front of a web cam or as defined by start and end points from a video, to look up its meaning and other related information in a multimedia sign language dictionary. A closely related application is the development of a sign-bank system to aid linguists with the task of preparing annotations for sign language video. The user identifies a segment in the input video sequence and the sign-bank system retrieves items from an annotated sign language dataset and in doing so allows the relevant linguistic attributes of the retrieved signs to be imported by the user.

Linguistically motivated probabilistic models (e.g., HMM, DBN, CRF) have shown substantial promise towards developing data-driven algorithms for spoken language recognition. However, linguistic properties have so far only been used to a very limited extent in the development of sign language recognition methods. One unique aspect of sign language is the way in which several articulatory channels (as detailed in the next section) are used

- **Palette for linguistic annotations [Neidle, 2007]**
  - Contains **85** handshape distinctions



Examples of handshapes observed in ASL video and their ground-truth labels from the handshape palette

- **Simple baseline for handshape recognition**
  - **30.4%** 1-nearest neighbor accuracy for 1,924 hand images

**Figure 1-2:** Examples illustrating challenges involved in identifying handshapes from hand images in ASL video. Signs in these video sequences were produced by a native signer.

together to convey meaning. Linguistic constraints govern the relationships among different articulatory features in visual-gestural productions that are deemed meaningful and valid in a signed language. Computer models that exploit linguistic constraints associated with different articulatory features can therefore enable SLR algorithms to yield a more linguistically plausible recognition result. This thesis focuses on the recognition of one specific articulatory component, handshapes. The properties of handshapes are relatively well-understood in terms of the features of hand configuration that convey essential distinctions among different signs as well as in terms of the constraints that are intrinsic to handshapes articulated in large classes of signs. Van der Kooij [Van der Kooij, 2002] and Whitworth [Whitworth, 2011] present an in-depth analysis of the properties of handshapes employed in signs.

We anticipate that handshape inference will be one of several computer vision components in a full-fledged SLR system. Figure 1.2 highlights a few of the challenges involved in developing a robust system for handshape recognition from sign language video. A baseline nearest neighbor handshape retrieval approach using isolated hand images yields 30.4% 1-nearest neighbor retrieval accuracy. Among the several options towards improving the handshape recognition rate, previous research in this area has not leveraged the linguistic properties that pertain to handshape articulation. For the handshape inference task, we formulate data-driven probabilistic models to leverage constraints on the allowable handshape relationships for the largest class of signs in American Sign Language (ASL). The models developed in this thesis, however, have more general applicability since the same types of principles could be applied to the recognition of handshapes in other signed languages, as well.

The availability of large corpora for both written language text and spoken language utterances has proven to be instrumental in developing state-of-the-art speech recognition systems. Unlike most spoken languages, signed languages do not have a standard, conventional written representation. Video corpora for signed languages annotated with linguistic information are therefore indispensable for developing SLR approaches. However, only a relatively small number of such corpora are currently available for sign language research. These datasets are also modest in size (especially with respect to the availability of productions from many different native sign language users) and often do not contain the necessary linguistic annotations. A corpus for ASL containing a large number of citation form signs produced by up to six native ASL users was therefore collected and annotated with the linguistic attributes necessary for training the proposed models for handshape inference. All aspects of the dataset preparation (recruiting native sign language users, eliciting signs from the participants, and preparing detailed annotations for the collected signs) involved very substantial contributions from sign language linguists.

## 1.1 Aims of this dissertation

In this thesis, we focus on the recognition of manual signs (i.e., words produced using the hands and arms). In particular, we limit our attention to one important manual component, handshape. The other manual components that are outside the scope of this thesis include: hand orientation (the direction vector perpendicular to the face of the palm), the hand location (points of contact with other parts of the body, or, its location in signing space with respect to the face, the other hand or the torso) and the hand movement trajectory.

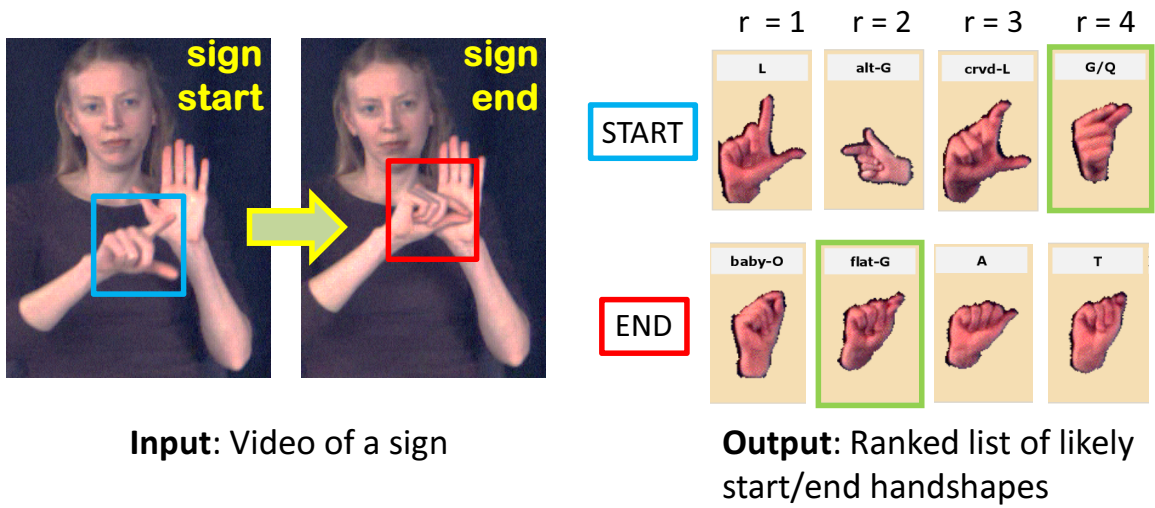
Handshapes represent the internal configurations of the hand (the degree of bending/extension of the skeletal joints in the palm and the different fingers). Only a finite number of handshape configurations convey linguistic distinctions in a given language<sup>1</sup>. Algorithms to identify handshape configurations observed in signing video are therefore an important component of SLR approaches.

Several factors have an adverse impact on the reliability of approaches for handshape recognition from a single image. Many handshapes differ from each other in the positions of one or more fingers (or the thumb) and hence are difficult to tell apart when observed by a camera from different viewing positions. Anthropometric differences as well as articulatory differences are observed in handshapes produced by different signers. The rapid movement of hands during signing produces motion blur in video. Hands also frequently occlude each other when signs are viewed from a particular viewpoint.

In this thesis, we aim to improve handshape recognition accuracy by exploiting several sources of regularity in handshapes produced within signs. In a large class of signs (monomorphemic lexical signs, the linguistic properties of these signs are discussed in the next chapter), the handshapes at the start and end temporal positions of the sign are essential for conveying linguistic distinctions. Signs in this class can be categorized as either one-handed or two-handed. In both types of signs, the changes in handshape configurations between the start and end positions of each sign are linguistically constrained. Furthermore,

---

<sup>1</sup>There is no general agreement regarding the precise number of handshapes in sign language, however, it is widely accepted that there are between 80 and 150 handshape forms that convey linguistic distinctions in ASL [Whitworth, 2011].



**Figure 1-3:** An example illustrating the handshape inference problem in monomorphemic lexical signs solved by the HSBN model formulated in this thesis. The signer’s right hand has been chosen here for the handshape inference problem to simplify this illustration.

in two-handed signs, there are linguistic dependencies between the handshapes of the left and right hands.

These constraints on handshapes can be represented probabilistically in a data-driven formulation that exploits the statistical properties of start and end handshape co-occurrences observed among a large number of signs in a sign language video corpus. In a similar fashion, the statistical properties of handshape variability in signs obtained from different signers can also be analyzed.

A formal problem statement for handshape inference in monomorphemic lexical signs can be summarized as follows. The handshape inference problem is illustrated with an example in Figure 1-3. The inputs given are a monocular video sequence depicting an isolated sign along with certain attributes of the query (these include, the start/end time-codes of the sign and a classification that specifies whether the sign is one-handed or two-handed). The desired outputs are the start/end handshape labels. In one-handed signs we need start/end handshape labels for the dominant hand whereas in two-handed signs we need the start/end handshape labels for both hands. A ranked order of the likely start/end handshape labels is desired as output of the handshape inference procedure in order to accommodate the



ambiguity inherent in determining handshape configurations from monocular video. A ranked ordering of the inferred handshape labels facilitates the integration of handshape inference results with the results obtained from other computer vision based recognition components in a SLR system towards the overall goals of producing a ranked list of sign language productions (i.e., labels for items contained in the vocabulary) for the query sign.

## 1.2 Overview of the proposed formulation for handshape inference

We propose the HandShapes Bayesian Network (HSBN) as a probabilistic representation towards addressing the handshape inference problem. The HSBN belongs to the class of Dynamic Bayesian Network models (DBNs). The HSBN’s model structure (i.e., the variables contained in the model and the probability distributions that relate the values adopted by these variables) is designed to incorporate linguistic properties that pertain to start and end handshapes in monomorphemic lexical signs. The HSBN employs hidden variables to account for inter- and intra-signer variability observed in handshape articulation. This allows certain handshapes to be obtained as different realizations of underlying hidden states.

The HSBN formulation is data-driven in that the parameters for probability distributions in the model as well as the representation for hidden variables in the model are estimated given a sign language video dataset annotated with linguistic attributes (these are outlined in the next section). The training of the HSBN model parameters is accomplished in a variational Bayes learning framework.

At query time, given parameters for the previously trained HSBN model, the posterior probabilities necessary for handshape inference can be computed in a closed form. The HSBN thus enables efficient algorithms for handshape inference towards producing a ranked list of handshape labels for handshapes contained in the query.

### 1.3 The lexicon video dataset for ASL

A corpus for ASL containing a large number of signs in the vocabulary produced by many sign language users is needed in order to develop our envisioned query-by-sign dictionary and sign-bank applications. Detailed linguistic annotations for the video sequences are necessary so that computer vision methods can be trained to make the same distinctions as those recognized by sign language users.

The ASL lexicon video dataset (ASLLVD) [Neidle et al., 2012b, Athitsos et al., 2008b] was developed at Boston University through an effort led by sign language linguists (under the direction of Carol Neidle) working in close collaboration with computer scientists. Linguists were chiefly responsible for recruiting native sign language users with diverse linguistic backgrounds, eliciting a large number of signs from the signers, developing the SignStream<sup>®</sup>3 application for conducting annotations of video sequences and, for the painstaking efforts required in preparing detailed annotations of linguistic attributes and articulatory features for the collected signs. The computer science contributions include the capture of high-speed time-synchronized videos from multiple viewpoints in a calibrated environment and the development of software to aid linguists in the task of verifying and ensuring consistency of linguistic annotations across  $\approx 10,000$  productions of signs contained in the dataset. Further details of the lexicon dataset will be presented in Chapter 4.

In the context of the handshape inference problem studied in this thesis, the lexicon dataset contains  $\approx 3000$  distinct monomorphemic lexical signs with examples of each sign produced by between one and six native sign language users providing a total of  $\approx 8500$  signs. This dataset is unique in that the signs

- (a) are grouped to ensure that each group of signs corresponds to a distinct item in the vocabulary thereby ensuring that the distinctions necessary for training computer models are delineated, and,
- (b) are annotated with several important linguistic attributes that include the start/end positions of signs in video, the start/end handshape labels and articulatory classifica-

tions denoting signs as one-handed/two-handed (and their sub-classes).

These annotations play a crucial role in the training and empirical evaluation of the HSBN formulation.

#### 1.4 Summary of contributions

The contributions of this thesis pertain to the development of the HandShapes Bayesian Network formulation for handshape inference and to the development of the lexicon video dataset for ASL.

- **The HandShapes Bayesian Network:**

The HSBN is formulated as a probabilistic approach to model linguistic properties and constraints that govern the allowable combinations of start/end handshapes in monomorphemic lexical signs. The HSBN model parameters are estimated given a dataset of signs annotated with linguistic attributes. By utilizing linguistic constraints during handshape recognition, the HSBN approach narrows the set of candidate labels for the observed handshapes in a given sign and thereby enables the recognition algorithm to produce a more linguistically plausible set of handshape labels.

The HSBN seeks to represent the properties of handshape articulation that hold in *general* for monomorphemic lexical signs. Robust models can therefore be trained even with modest dataset sizes containing a relatively small number of examples for several items in the vocabulary. Sign-specific models that have traditionally been used for SLR perform poorly on the signer-independent recognition task due to the small number of examples that are typically available for each item in the vocabulary.

The HSBN utilizes a hidden (hidden) variable layer to accommodate inter- and intra-person variations in handshape articulation. The handshapes observed in signs can therefore be modeled as different realizations of these hidden variables.

To impart additional robustness to anthropometric variations, we develop an efficient algorithm to perform non-rigid image alignment for handshape image pairs. This aids

in improving the set of candidate handshape labels that are obtained as potential matches to the handshapes observed in a query sign.

The HSBN has been evaluated for the person-independent handshape recognition task using a dataset containing a large number of distinct ASL signs. The HSBN demonstrates improved recognition accuracy when compared to an approach that recognizes start and end handshapes independently (more details about the evaluation method and accuracies obtained are described in the next section).

- **Lexicon video dataset (ASLLVD):**

The lexicon dataset provides a large collection of ASL signs annotated with linguistic attributes to enable the development of data-driven probabilistic models for SLR. Even though the focus in this thesis is handshape recognition, we anticipate that the lexicon dataset can prove to be an important resource in facilitating research in other aspects of SLR, as well as linguistic research on ASL. The lexicon dataset is also essential to enable progress towards our envisioned SLR applications: the query-by-sign dictionary and sign-bank systems.

The lexicon dataset was developed through a collaborative effort involving a large team of linguists and computer scientists. The contributions of the research in this thesis towards this project was the development of the Lexicon Viewer and Verification Tool (LVVT), a software application to organize, verify and ensure consistency of linguistic annotations across several thousand signs contained in the dataset. The LVVT provides linguists with the functionality necessary to efficiently detect and annotate fine-grained distinctions among different productions of signs. These distinctions are essential in general for training computer vision methods for SLR. These distinctions are specifically leveraged in this thesis to train probabilistic models for handshape inference.

## 1.5 Summary of results

The HSBN was evaluated using signs contained in the ASLLVD collection. A training set is used to learn model parameters in the HSBN using the variational Bayes learning approach. The training set contains 2,636 monomorphemic lexical signs produced by 5 native signers for a total of 6,958 examples. A sequestered subset of signs produced by a signer who is not part of the training set is used to evaluate the handshape inference accuracy. The test set contains 577 signs from one signer providing a total of 646 examples.

Using the HSBN to perform joint inference of start/end handshape labels improves signer-independent rank-1 handshape recognition accuracy from 30.4% (for the baseline simple nearest neighbor based handshape recognition method) to 44%. This accuracy may appear low, but it is a significant improvement given the large number of handshape classes (85 labels) with relatively small differences in handshape configuration.

## 1.6 Thesis roadmap

The thesis is organized as follows. Background concerning the linguistic concepts that motivate the handshape inference formulation developed in this thesis is presented in Chapter 2. An overview of the ASL Lexicon Video Dataset is given in Chapter 4. Previous work on computer vision methods that have addressed the problem of recovering hand configurations from video for both sign language applications as well as in non-signed gestures are discussed in Chapter 3. The HSBN representation for the handshape inference problem is formulated in Chapter 5. The equations and algorithm for training the HSBN model are derived in Chapters 6 and 7. The proposed approach for non-rigid image alignment to extract nearest neighbors for query handshape images that are used to compute observation likelihoods in the HSBN is described in Chapter 8. The performance of the proposed handshape inference approach is evaluated in Chapter 10. A summary of the contributions and potential future extensions of this work are presented in Chapter 11.

## 1.7 Publications reporting research conducted in this thesis

A preliminary version of the HSBN formulation for one-handed signs was reported in [Thangali et al., 2011]. A previous version of the image alignment algorithm was evaluated for the hand detection task in [Thangali and Sclaroff, 2009]. The ASLLVD dataset was described in [Neidle et al., 2012b, Athitsos et al., 2008b]. Additional details of the dataset are provided on the ASLLVD webpage [Neidle et al., 2012a].

## Chapter 2

# Background about American Sign Language (ASL)

This chapter summarizes the linguistic background needed to motivate the HSNB formulation for the handshape inference problem.

### 2.1 Introduction to sign language

In this brief introduction, we provide an overview of linguistic properties of signed languages that are relevant to this research. For a general introduction to sign language including an overview of its linguistic properties and the internal mechanisms governing the composition of signs, see [Valli and Lucas, 2000, Brentari, 1998, Van der Kooij, 2002].

#### 2.1.1 Challenges that arise in developing linguistic models of sign language

Sign languages are comparable in richness, structure, and complexity to spoken languages. Signs are the analogs of words in the visual-gestural modality. Signs may be morphologically inflected to incorporate information about, for example, aspect or agreement; and the realization of a sign can be affected by adjacent signs, giving rise to co-articulation effects.

Unlike most spoken languages, sign languages generally do not have a standard, conventional written form. Signed and spoken languages in a given geographic region bear relatively little relationship (there are however phenomena resulting from language contact). Written language texts are hence not available for the development of sign language models. Labor intensive linguistic analysis and annotation of sign language video sequences is often the only viable means of accruing a sufficient amount of data to enable the development of (theoretical or computer-based) models to represent linguistic processes involved in sign language. Our attention is focused on developing computer models for articulatory

processes that are general to a large class of signs and we can thus circumvent some of the difficulties posed by the relatively small sizes of sign language datasets that are currently available.

As with spoken language, dialectal and idiolectal differences as well as naturally occurring variations in articulation are found in sign language productions from different users. Such variations must be taken into account in computer methods developed for person-invariant SLR systems. In this research we adopt a data-driven approach to formulate probabilistic models that account for sign-independent handshape variations attested in a large class of signs.

## 2.2 Linguistic organization of signed languages

Signs are produced by articulations of the hands and arms. Non-manual expressions, i.e., expressions of the face and upper body occurring in parallel to manual signing, also convey important linguistic information. In this section we will describe the internal composition of signs.

### 2.2.1 Units of meaning and articulatory units conveying linguistic distinctions in (signed / spoken) language

The basic units of meaning (*morphemes*) in both spoken and signed languages are made up of articulatory, discriminatory units called *phonemes*. (Many linguists use this term even for signed languages.) In spoken languages, these discriminatory units are articulations produced through the vocal tract and perceived auditorily. The discriminatory articulatory units in sign language are perceived visually. Hand shapes, orientations, and locations within the signing space, as well as, movement type (and in some cases non-manual expressions of the face or upper body) are among the components for which distinctive values can differentiate meanings among signs.



### 2.2.2 The internal structures of morphemes

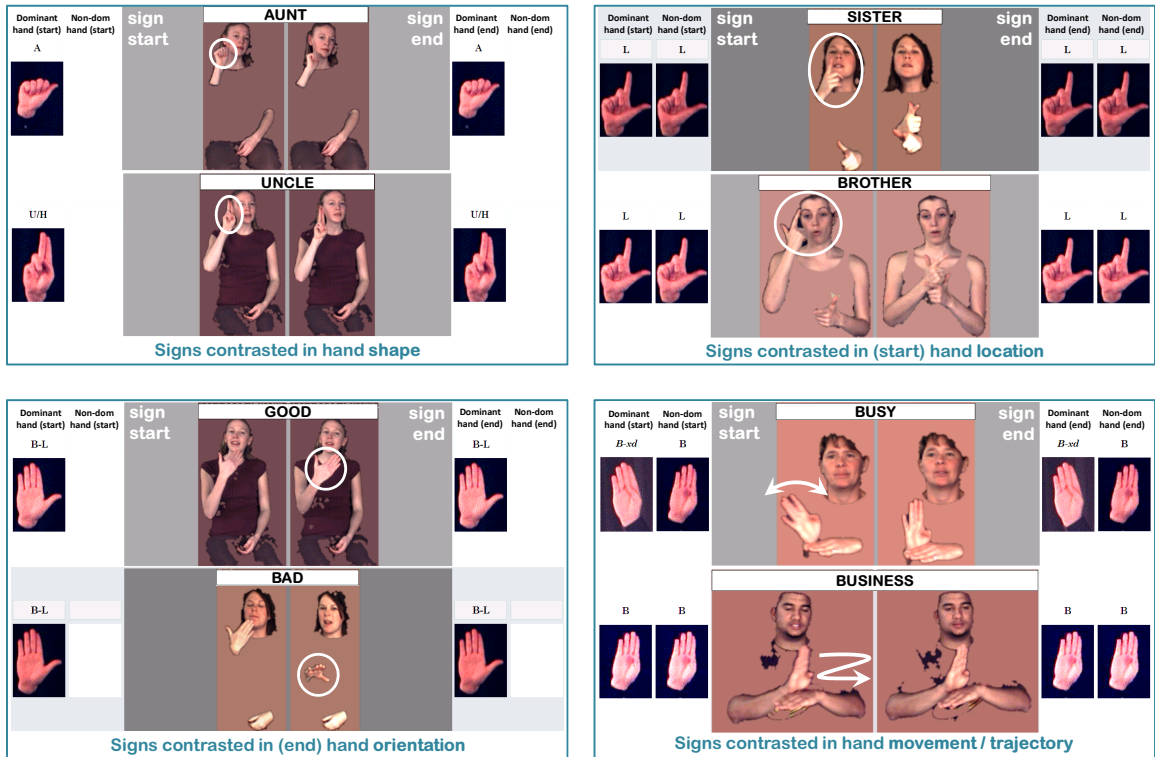
The basic principles according to which phonemes combine to form morphemes is essentially the same within and across spoken languages: phonemes combine linearly (i.e., sequentially) although non-linear phenomena (e.g., co-articulation effects) also play a very significant role in speech production.

In signed languages, however, there are different morphological classes that are governed by somewhat different compositional principles. This poses a problem for computer-based sign recognition that does not exist for spoken languages: linguistically based models used for computer-based sign recognition must be appropriate for the specific type of sign involved. The rest of this discussion will focus specifically on ASL. The types of distinctions found in ASL are relevant in other sign languages, as well.

### 2.2.3 Morphological subclasses of signs in ASL

A typology of signs in ASL has been described by various linguists. The essential distinctions in ASL are outlined below (Brentari [Brentari, 1998] presents an in-depth discussion on this topic). In particular, the focus for the research in this dissertation will be on “lexical signs”, specifically, monomorphemic lexical signs.

- The subclasses of signs in ASL that will not be studied in this dissertation include,
  - *Fingerspelled signs*: Fingerspelling is often used for proper nouns or borrowings from spoken language and consists of a sequence of handshapes from the manual alphabet that are used to spell out letters in an English word.
  - *Loan signs*: Loan signs are a class of signs that result from borrowing from other linguistic sources. Many loan signs originated as fingerspelled signs but have undergone a process of lexicalization. Often a characteristic hand movement is involved in addition to the handshape articulation of the letters.
  - *Classifier constructions*: The types of movements allowed in classifier constructions are far greater than in other types of signs. In some classifier constructions,

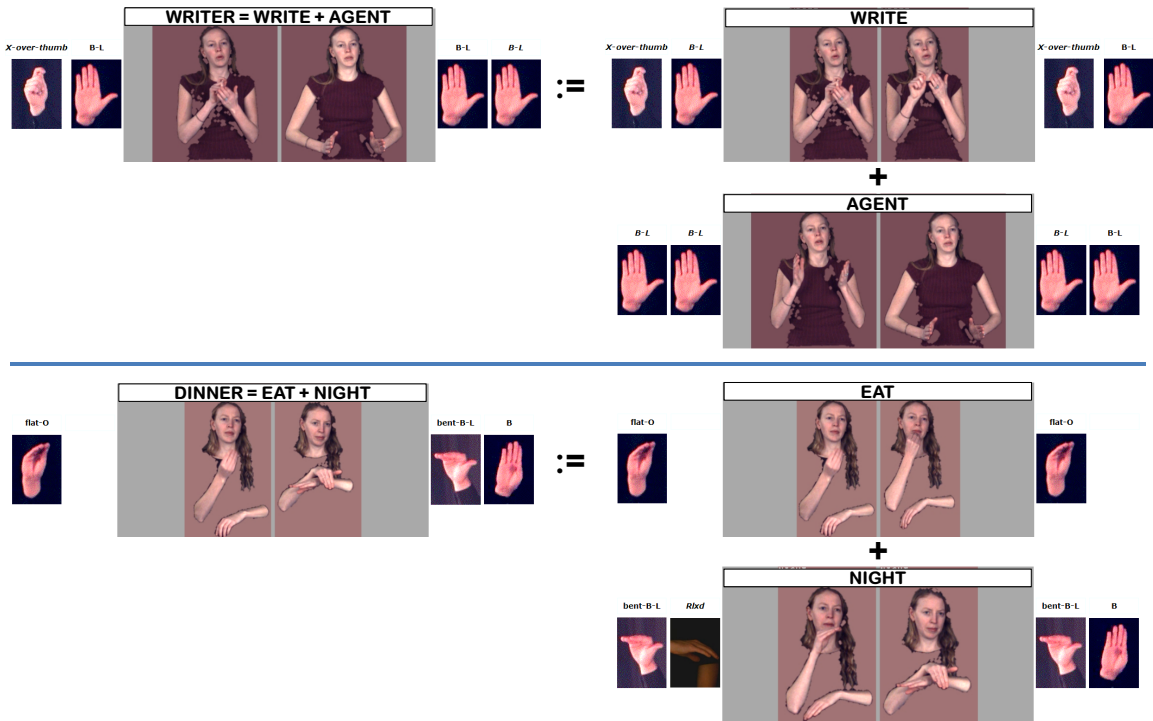


**Figure 2.1:** Examples for pairs of signs minimally distinguished by hand-shape, hand location, hand orientation, or, hand movement trajectory.

for example, a handshape representative of a class (e.g., vehicle, human) is combined with a movement expressing verb/action/spatial-relationship/manner.

- Lexical signs are the focus of this research. Morphemes in lexical signs are built up through linguistically constrained choices of handshape, orientation, location, and movement, which occur simultaneously. Articulatory elements that compose and distinguish morphemes in lexical signs are often referred to as phonemic because they serve a role analogous to phonemes in spoken language.

Examples of articulatory contrast conveyed by different features are illustrated in Figure 2.1 using pairs of lexical signs minimally distinguished by a change in one of the parameters. **AUNT** and **UNCLE** are distinguished by the handshapes 'A' and 'U' used on the dominant hand. **SISTER** and **BROTHER** are distinguished by the location of the dominant hand at the start of the sign (the hand is close to the chin in the former



**Figure 2-2:** Examples of compound (or, polymorphemic) signs.

while it is close to the cheek/forehead in the latter). GOOD and BAD are distinguished in the orientation of the palm of the dominant hand at the end point of the sign (the palm of the dominant hand faces the signer and the floor respectively) and also potentially in their facial expressions. BUSY and BUSINESS are distinguished in the movement patterns of the dominant hand: in the sign BUSY, the wrist of the dominant hand rotates; whereas, in the sign BUSINESS the wrist translates along the base arm.

- As in spoken languages, it is also possible to have signs composed of more than one morpheme. The examples, WRITER = WRITE + AGENT, and, DINNER = EAT + NIGHT, are illustrated in Figure 2-2. The morphemes that combine to form the compound signs in these examples are lexical signs. However, morphemes from other classes of signs can also appear in compound forms. Compounds are particularly interesting, because of the co-articulation effects observed at morpheme boundaries.

Even though such co-articulatory influences are, to some extent, predictable, they also introduce recognition challenges. Co-articulation often ‘blurs’ the boundary between the two segments: The end state of the previous morpheme and the start state of the following morpheme are modified in relation to the forms that they would take in isolated productions. Although co-articulatory effects are also observed at word boundaries in continuous signing sequences, the magnitude of these effects tends to be greater within compounds than between signs. Compound signs provide a controlled linguistic environment to facilitate the study of co-articulatory patterns.

The dataset collected for this research contains  $\approx 350$  instances of compound signs. Their analysis could yield valuable insights for both linguistic analysis and computer-based modeling of co-articulatory phenomena. We envision that the handshape inference approach developed in this dissertation for monomorphemic lexical signs can be extended in future work to model co-articulatory phenomena in compound signs.

## **2.3 Handshapes in monomorphemic lexical signs**

In this research we focus on handshape, an important phonological element in sign language. We start with an overview of the different systems that have been developed by linguists for representing handshape configurations in signs. An appropriate representation of handshapes plays an important role in preparing annotations for sign language video. These annotations in-turn facilitate the development of data-driven probabilistic models for representing different linguistic properties that pertain to handshape articulation in signs. Of particular interest to us are the constraints that govern the combinations of handshapes in monomorphemic lexical signs. Also of interest are the patterns of variation attested in the production of handshapes. These two topics are discussed in subsequent sections below.

### **2.3.1 Handshape representation**

There is no general consensus regarding the set of basic handshapes in ASL (and also in other signed languages). Even though the different systems proposed for handshape

representation differ in the granularity of the features that serve as a basis for labeling handshapes, the notion of ‘selected fingers’ plays an important role in encapsulating the salient properties of a given handshape configuration. Selected fingers are a subset of fingers that are salient in the articulation of a specific handshape [Liddell and Johnson, 1995, Brentari, 1998, Van der Kooij, 2002, Whitworth, 2011]. Examples of selected fingers within different handshapes and the different configurations of selected fingers in these handshapes are described further below.

Approaches for handshape representation can be broadly classified as follows:

- **Representations that encode the joint positions for each of the different (selected) fingers**

These systems use the following parameters to represent each handshape configuration

- The subset of fingers that are selected (or, has salient properties in the articulation of a specific handshape).
- The different degrees of bending/extension at the base and non-base joint angles of the fingers. Liddell and Johnson [Liddell and Johnson, 1995] have suggested four states  $\{closed, hooked, extended, flattened\}$  along with a symbol to denote a degree of flexibility in the muscle action (i.e., a relaxation in the encoded amplitude of folding/extension at a finger joint).
- The degree of spreading between different fingers (denoting abduction / adduction for the selected fingers).
- The different positions of the thumb with respect to the palm. Van der Kooij [Van der Kooij, 2002] suggests the features  $\{crossed, opposed, adducted, and, extended\}$  along with an aperture feature (closed or open) that is useful when the thumb is in an opposed configuration to denote whether the thumb is in contact with the selected fingers.

An explicit representation for hand configuration allows for the precise encoding of a wide range of handshape configurations albeit at the expense of significantly ex-

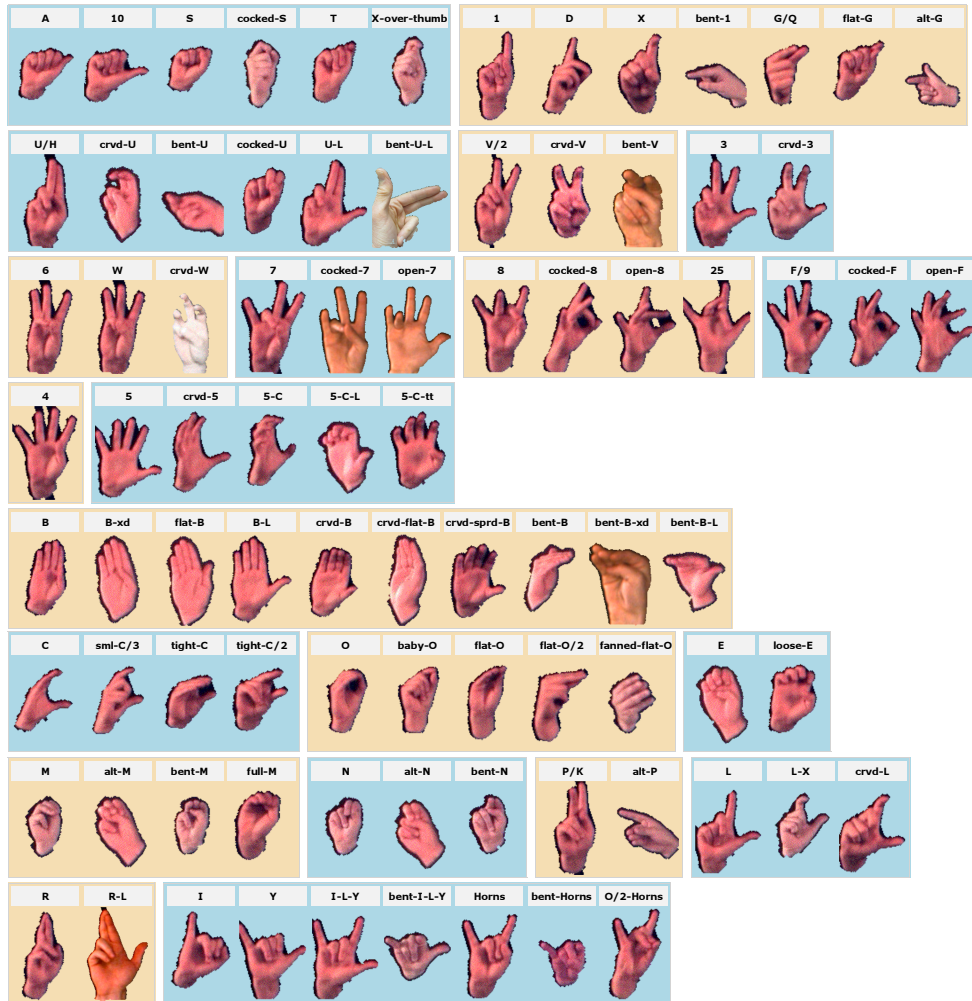
panding the state-space of representable handshape configurations. This substantially increases the model complexity for handshape inference and the annotation effort involved in labeling the ground-truth. The linguistic significance of many of these handshape configurations is also difficult to ascertain (“... these models may offer inventories of very detailed categories with numerous form elements often without addressing their distinctiveness” [Demey and Van der Kooij, 2008]).

- **Representations that are based on wholistic configurations of the hand**

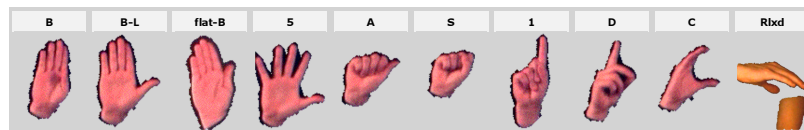
In approaches that enumerate wholistic configurations of the hand, an inventory of handshape forms is determined through an analysis of the different handshape configurations observed in sign language video datasets. Handshapes that are attested as producing articulatory contrast or those attested as conveying certain salient linguistic properties in signs are included in the inventory.

Many of the challenges that arise with employing an explicit encoding of hand configurations are circumvented to some extent in the handshape inventory approach. The latter approach sacrifices some precision in the transcription of specific handshape forms to provide a more compact representation. The smaller number of handshape distinctions facilitates handshape annotation in sign language video sequences as well as the development of computer models to represent the properties of handshape combinations in signs. We therefore adopt an inventory-based representation of handshapes in this research.

We utilize an inventory containing 85 handshapes to denote handshape configurations in this work Figure 2.3. Handshapes in this inventory were selected by linguists through an analysis of approximately 10,000 isolated (citation form) signs in the ASLLVD [Neidle et al., 2012a] and 10,500 utterances within continuous signing sequences from the ASLLRP [Neidle, 2013] video collection. Handshapes in the inventory are grouped based on similarity into different subsets. These groups of handshapes were created to aid the organization of handshape labels in the inventory and do not necessarily reflect linguistic affinity among



(a) Set of all handshape labels for ASL annotations in [Neidle, 2007].



(b) Unmarked handshapes in ASL.

**Figure 2-3:** The 85 handshapes in ASL labeled according to annotation conventions in [Neidle, 2007]: (a) The dominant signing hand can take any handshape from this set; (b) The handshape on the nondominant hand, when it differs from that of the dominant hand in a two-handed sign, is constrained to belong to the set of *unmarked* handshapes. Video sequences displaying multiple views of each of these handshapes in motion are available in [Neidle, 2011].

different handshape configurations. The distinctions drawn in handshape configuration include: different degrees of folding/extension of the selected fingers (e.g., {U, bent-U, curved-U, cocked-U}) and different positions of the thumb within the same basic handshape (e.g., {B, B-xd, flat-B, B-L}), different degrees of spread between the selected fingers (e.g., {crvd-B, crvd-sprd-B}) and different degrees of aperture of the hand (e.g., {O, flat-O, fanned-flat-O}).

### 2.3.2 Linguistic constraints governing handshape articulation in monomorphemic lexical signs

In monomorphemic lexical signs, handshapes play an important role in distinguishing signs, and the most linguistically informative portions with respect to handshapes are observed at the *start* and *end* points of signs (on either the dominant hand in one-handed signs or on both hands in two-handed signs). With the exception of a small number of signs that include explicit finger movements (e.g., wiggling, waving or rubbing of fingers), the intermediate handshapes are often predictable given the start and end handshapes.

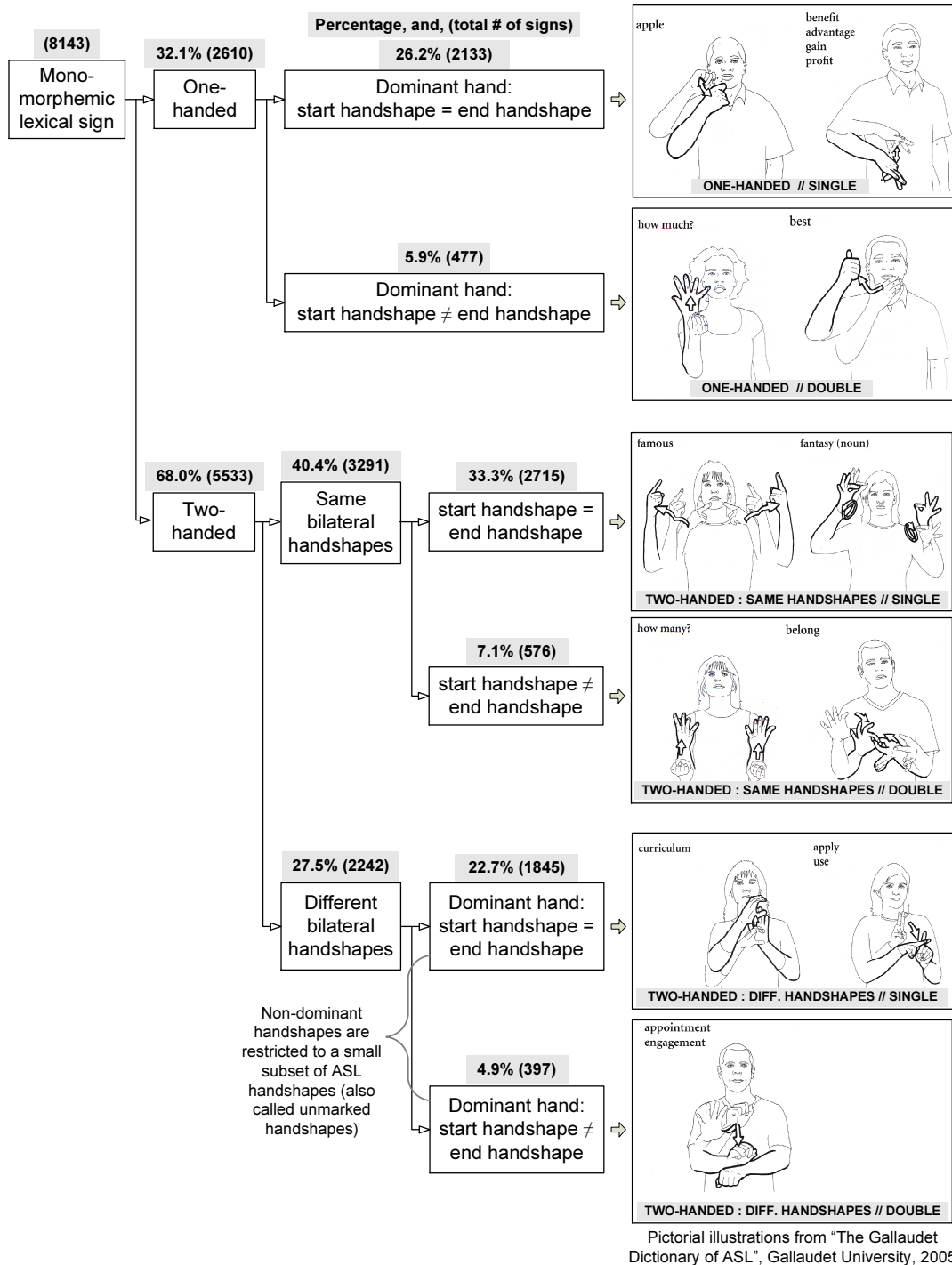
Battison's taxonomy [Battison, 2000] for constraints on handshape articulation in monomorphemic lexical signs is illustrated in Figure 2.4. The constraints are broadly categorized into those that relate the start and end handshapes for a given hand, and, those that relate the start/end shapes used by the two hands in 2-handed signs:

- **Relationships between start and end handshapes for a given hand**

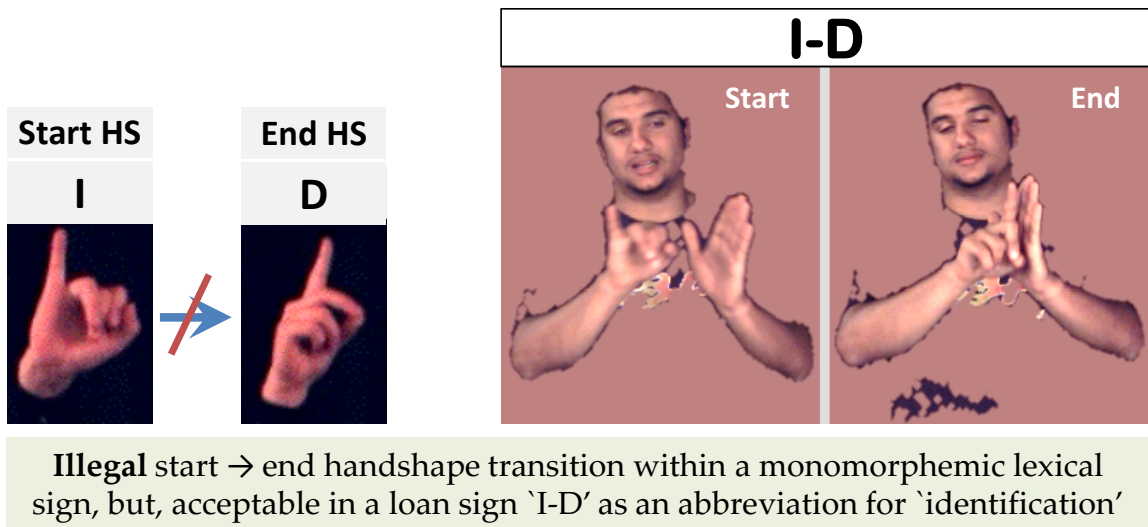
The handshape used at the start of a sign constrains the set of handshapes that can appear as end handshapes on the same hand. This is because, in general, only the *selected fingers* can exhibit a change in configuration while the unselected fingers normally do not change their configuration.

The significant types of changes in handshape configuration attested between the start and end points of signs include: the bending/extension of the base (metacarpal-phalangeal or MCP) and non-base (proximal/distal inter-phalangeal or DIP/PIP) joints for the selected fingers, changes in the spread (abduction/adduction) of the selected fingers, and, closing/opening changes in the aperture of the palm and/or the





**Figure 2-4:** A taxonomy of constraints on handshapes for monomorphemic lexical signs in ASL [Battison, 2000]. The percentages (and total numbers of signs) in the ASLLVD collection corresponding to each constraint are also shown.



**Figure 2-5:** Changes in hand configuration within monomorphemic lexical signs are constrained to involve either closing or opening of the hand (i.e., the folding/unfolding of the palm and a selected subset of fingers). A sign from the class of *loan signs* that violates this constraint is shown here.

hand.

An example of an illegal change in handshape configuration within monomorphemic lexical signs is illustrated in Figure 2-5. The change in handshape  $I \Rightarrow D$  involves the simultaneous closing of the index finger and extension of the little finger. Such a transformation is disallowed in monomorphemic lexical signs because the selected set of fingers is modified in order to transition between these two handshapes. However, such a sequence is produced in a *loan sign* I-D realized as an abbreviation of the word 'identification'.

- **Relationships between start/end handshapes used by the two hands in 2-handed signs**

Monomorphemic lexical signs are classified as one-handed or two-handed based on whether one or both hands normally participate in the production of a sign (although there is some variability in the number of hands used for a particular sign). For some two-handed signs, there is asymmetry in the use of the two hands in sign production. The dominant hand (the hand used for one-handed signs, or, the hand that carries the

most salient information in a two-handed sign) is usually determined on the basis of a signer’s bilateral preference for motor tasks. Sometimes in conversation/narrative, however, the signer may switch dominance in signing.

We use the term `two-handed:same handshapes` in this thesis to identify signs that exhibit bilateral symmetry in handshapes on the two hands. We ignore global hand movement in defining this class. The `two-handed:same handshapes` signs thus include signs in the ‘Type 1’ and ‘Type 2’ categories identified by Battison [Battison, 2000] (these two classes in Battison’s notation correspond to signs that exhibit bilateral symmetry in only handshape, and, bilateral symmetry in both handshape and hand movement respectively).

In two-handed signs where the two hands take different handshapes (referred to as `two-handed:different handshapes`), the handshape on the non-dominant hand is restricted to a small subset of unmarked handshapes (Figure 2.3 lists the set of unmarked handshapes in ASL). Furthermore, in such cases the non-dominant hand does not exhibit any change in handshape configuration between the start and end points of the sign.

One-handed and two-handed signs can be further classified based on whether the start and end handshapes are the same or different (‘single’ and ‘double’ in Battison’s terminology [Battison, 2000]). When the start and end handshapes differ, the change in handshape is constrained to involve either the closing or opening of the hand as noted in the previous section (Section 2.3).

### **2.3.3 Handshape variation in monomorphemic lexical signs**

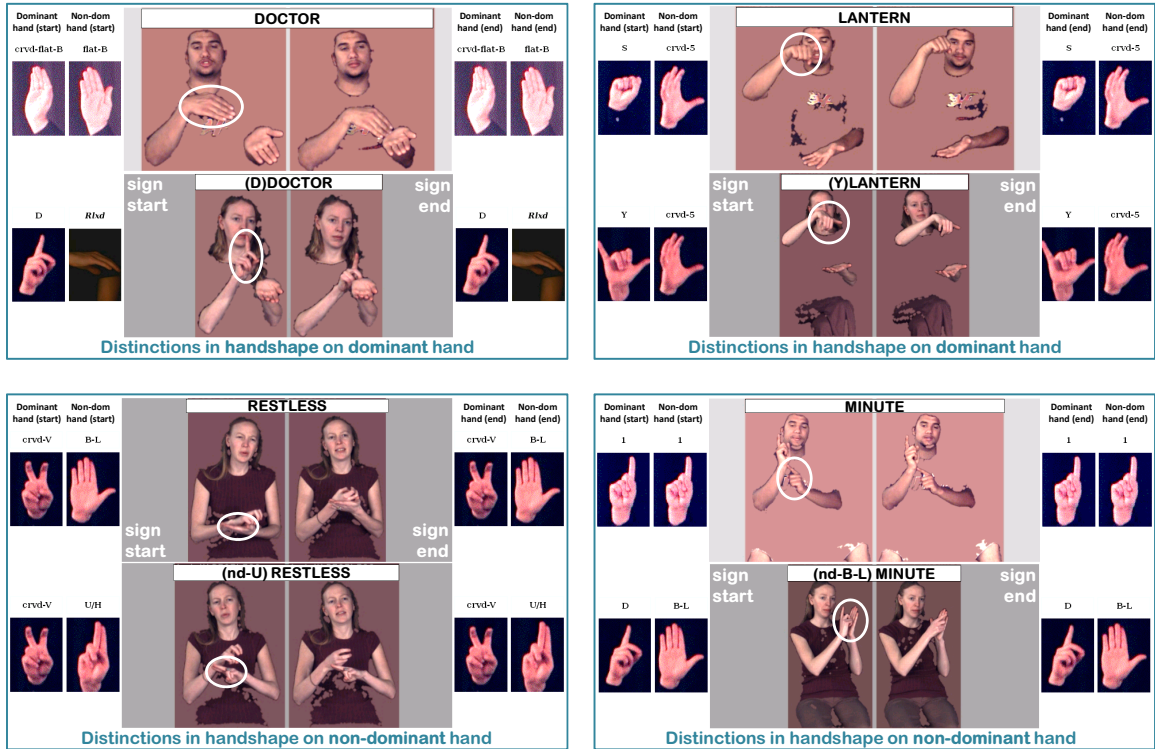
Handshape variation in sign language, in general, is less well-studied than the topic of handshape representation. Israel and Sandler [Israel and Sandler, 2009] review previous literature (e.g., [Battison et al., 1975, Van der Kooij, 2002]) analyzing handshape variations. Bayley et al. [Bayley et al., 2002] have studied the patterns of variation in the ‘1’ handshape for the sign DEAF. They find evidence for variations conditioned by factors such as grammatical function, assimilation effects from features of the preceding and following segments,

along with a range of social factors including age, regional origin and language background. Van der Kooij [Van der Kooij, 2002] presents an in-depth discussion of the different types of variations observed in handshapes. She describes phonological environments where variations in the thumb position (crossed, flat, opposed or extended) are observed and concludes that the non-extended thumb positions are often linguistically non-informative. She also finds evidence for non-distinctiveness of certain features of the selected fingers such as the degree of flexion of the base joint, the degree of flexion of the non-base joints and the degree of spreading of the fingers. Other features that can also demonstrate variability are the positions of the unselected fingers and the aperture of the hand.

The general consensus among researchers studying sign language is that a better picture for the phenomenon of variations will emerge from the analysis of large sign language corpora containing examples from many native speakers for each lexical item in the vocabulary. To narrow the focus of our discussion we restrict our attention to the properties of handshape variation attested within monomorphemic lexical signs. Furthermore, we are interested in analyzing variations in the productions of citation-form signs. We do not therefore consider assimilation effects due to either preceding or following sign segments as seen, for example, in compound signs and in continuous signing sequences.

In order to facilitate the analysis of handshape variation, we broadly classify the attested handshape variations into two classes: ‘alternations in handshape that are sign-specific (lexical variations)’, and, ‘alternations in handshape that are produced as a result of general language processes’.

(†) Variations in handshape that are attributable to general phonological principles are of particular interest in developing the HSBN model for the handshape inference task, and for purposes of this project, we have generally assumed that the handshape variations that are attested across some set of different signs result from application of such general phonological processes. However, it should be pointed out that from the limited data used in the current study (this dataset is described in more detail in Chapter 4), we cannot rule out the possibility that some of this variation may in fact, result from idiolectal or dialectal



**Figure 2-6:** Examples of pairs of signs exhibiting sign-specific variations in handshape on the dominant or non-dominant hand.

differences among signers in the phonological form used for specific signs. It is hoped that such issues can be sorted out in future linguistic research, but at present, we point out that the present approach is biased in favor of interpreting such patterns of handshape variation across multiple signs as phonological in nature. The anticipation is that future refinements in the linguistic analysis that may be achieved by work with larger datasets and queries of native signers may yield improvements in the statistical modeling.

- **Sign-specific variations in handshape**

Sign-specific variations are tightly linked either to a specific sign (i.e., a lexical item in the vocabulary) or to a small group of signs. The patterns of variation in this class are not generally attested among many signs in the vocabulary. We use the term ‘lexical variants’ in this discussion to refer to the different versions of a sign produced with sign-specific differences in articulation. The lexical variants of a given sign convey

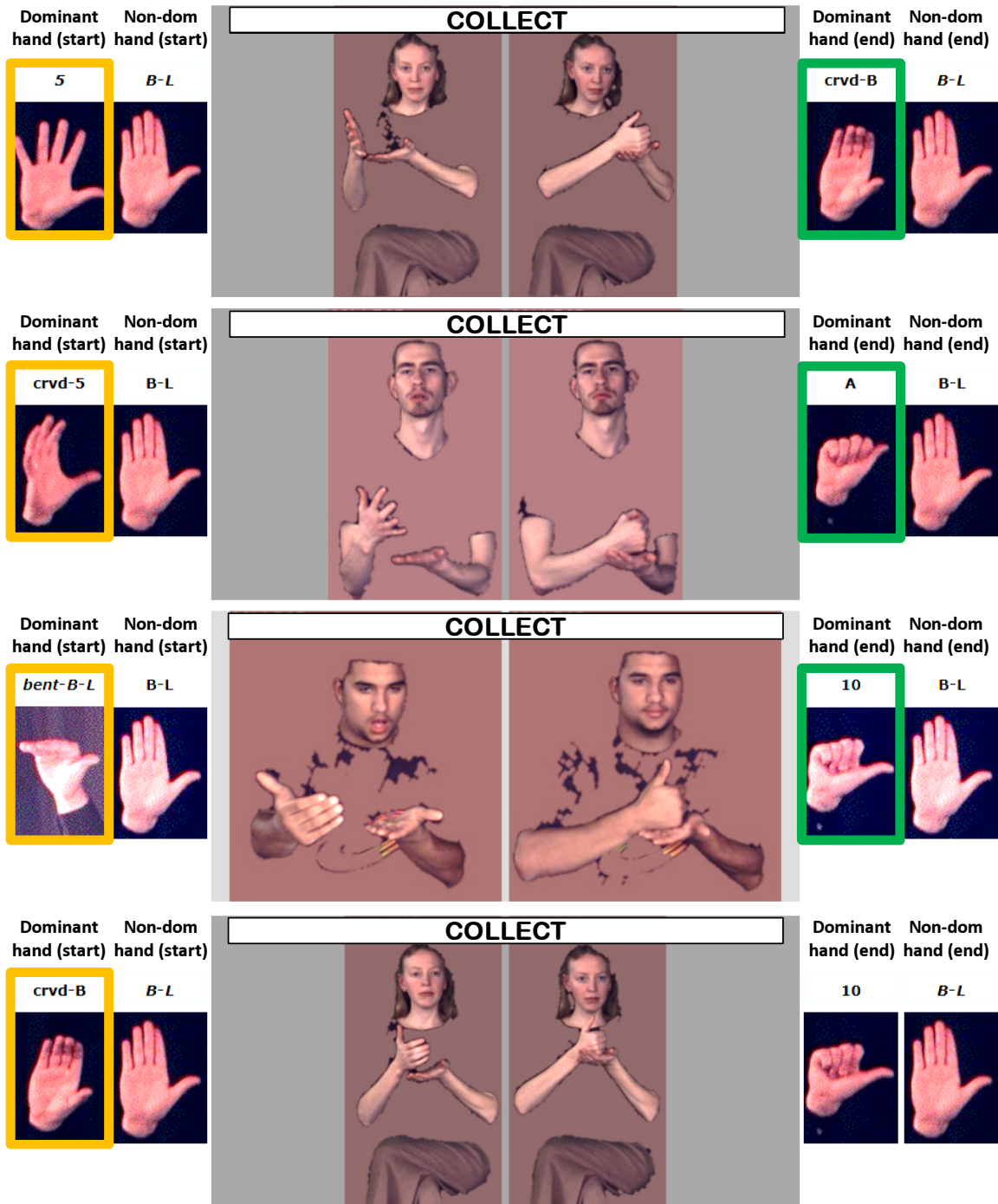
similar, but not necessarily identical, meaning and/or linguistic interpretation. Lexical variants of any given sign are treated as distinct signs in this research.

Examples of lexical variation in handshape are displayed in Figure 2-6. Lexical variants are assigned distinct gloss labels (sometimes but not always incorporating information about the distinguishing handshape for a given variant). {DOCTOR, (D)DOCTOR}, {LANTERN, (Y)LANTERN} differ in the handshapes articulated on the dominant hand, whereas {RESTLESS, (nd-U)RESTLESS}, and, {MINUTE, (nd-B-L)MINUTE} differ in the handshapes articulated on the non-dominant hand.

- **Variations in handshape that are produced as a result of general language processes**

The patterns of handshape variation in this class are hypothesized to reflect general phonological processes (but see the disclaimer, †, above). These patterns of variation are observed among the productions of many different signs in the language. We employ the term ‘phonological variation’ to refer to variations in this class. Phonological variations do not typically modify the linguistic interpretation of a given sign. Examples of phonological variations in the sign COLLECT are illustrated in Figure 2-7. Handshapes among different phonological variants are often closely related in terms of the underlying hand configurations.

We focus in this research on modeling the patterns of phonological variation in handshape that are attested among multiple productions of different monomorphemic lexical signs. We leverage annotations for start/end handshapes prepared by linguists for signs contained in the lexicon dataset (Chapter 4) for learning the properties of handshape variation. The lexicon dataset contains isolated productions of monomorphemic lexical signs organized into groups of distinct sign-variants. This grouping allows us to accrue the patterns of handshape variation using a data-driven approach to estimate the parameters of probability distributions that represent phonological variations within the HSBN model. Our aim with the HSBN is to represent the properties of



**Figure 2-7:** Examples of phonological variation in handshape. Variations in this class are not specific to a particular lexical item.



signer-independent variations in handshape and therefore person-specific preferences for certain handshapes are not modelled in our formulation.

## 2.4 Summary

Signed languages are analyzed by linguists as possessing properties parallel to those of spoken languages. However, words in spoken language are constructed from articulatory units that are combined essentially linearly (i.e., sequentially, although not without significant co-articulation effects), whereas different morphological classes of signs in signed languages differ in the compositional principles by which the units of meaning are constructed from articulatory units. The analysis of the internal composition of signs is therefore specific to signs in each class. For this brief introduction we restrict our attention to the class of lexical signs in ASL. Units of meaning, or morphemes, in lexical signs can be analyzed as being composed of linguistically constrained choices of different shapes, orientations, locations of the hand as well as movement types of the hands and arms. Non-manual articulations also play a role in certain signs. Focusing specifically on handshapes, many sign language researchers have identified particular hand configurations that are used distinctively for differentiating signs. In this research we utilize a palette comprising 85 handshapes developed by linguists for the purposes of preparing annotations for ASL video sequences.

The allowable combinations of different start/end handshapes on a given hand as well as on the dominant and non-dominant hands in monomorphemic lexical signs are governed by linguistic constraints that apply in general to monomorphemic lexical signs. Furthermore, certain types of variation in the articulation of handshapes in monomorphemic lexical signs can be analyzed as resulting from general language processes.

The constraints on combinations of different handshapes as well as the patterns of phonological variation in handshape are modeled using the HSBN representation in this thesis. A data-driven learning approach leveraging a large corpus of signs annotated with linguistic attributes is employed for the purposes of estimating the model parameters. The collection and development of the lexicon dataset arose as one of the contributions of a



research project aimed at furthering the development of a query-by-sign lexical lookup system for an ASL dictionary. This dataset is described in more detail in Chapter 4.

## Chapter 3

### Related Work

We review approaches for modeling and recognition of manual articulations. Approaches for human gesture recognition and retrieval are frequently formulated in a trajectory analysis framework wherein temporal sequences of hand locations, the positions and orientations of the upper and lower arms are augmented with features describing the appearance of handshapes. In the context of the handshape recognition research conducted here, we place particular emphasis on approaches that either explicitly model or exploit features of hand articulation. Since linguistically annotated sign language datasets that are currently available do not include 3D and/or depth information, we focus our attention here on video analysis based approaches.

We start with an overview contrasting modeling based approaches with nearest neighbor retrieval approaches for incorporating spatio-temporal properties of human gestures. Outside of SLR there is a large body of computer vision research on tracking finger articulations in video sequences; we describe some of these previous approaches in relation to the current work. In the SLR context, several approaches for handshape recognition from isolated hand images have also been developed.

The Hidden Markov Model (HMM) and Conditional Random Field (CRF) models are two frequently used probabilistic formulations for representing the sequential properties of articulation in human gestures. HMMs for different gesture classes are often trained independently. The algorithms for training the HMMs and for performing inference given a query sign are thus efficient. HMMs, however, are not trained explicitly to distinguish one gesture from another. To overcome this drawback, different variants of the CRF [Lafferty et al., 2001] formulation have been proposed to jointly model all the gesture classes in

the vocabulary in a discriminative fashion. Dynamic CRF [Sutton et al., 2004], Hidden CRF [Wang et al., 2006] and the Latent Dynamic CRF [Morency et al., 2007] are a few examples. The LDCRF, for example, models the transitions between different gestures (thus capturing extrinsic dynamics), and also incorporates hidden state variables to model the internal sub-structure of gestures. Promising results for sign-spotting using CRF’s are demonstrated in [Yang et al., 2009] (in a sign-spotting problem the start/end positions for an input sign’s occurrences in a continuous signing query sequence need to be determined). Nearest neighbor (NN) retrieval methods compute a similarity score between pairs of signs (for instance, a query and a database sign) based on the trajectory and appearance features. Approaches for nearest neighbor retrieval typically compute a spatio-temporal alignment by employing the Dynamic Time Warping (DTW) algorithm to account for some of the variation in the locations of hands and for variations in the speed of articulation [Dreuw et al., 2006, Alon et al., 2009].

HMM and CRF based modeling methods assimilate information from multiple training examples for each item in the vocabulary and therefore provide improved robustness to variations in the articulation. Retrieval methods on the other hand are better suited to datasets that do not contain multiple examples for each class label or in datasets where a linguistic analysis to group examples into distinct lexical items is not yet available. HMM and nearest neighbor retrieval based approaches have been widely studied for SLR applications, we describe these works in greater detail in the subsequent sections. In both classes of approaches, however, phonological constraints that govern the different articulatory parameters in signs have so far not been fully exploited. Furthermore, signer-independent large vocabulary SLR presents a difficult challenge for computer vision approaches. In order to make progress towards both these goals, in formulating the HSBN model we leverage linguistic constraints that pertain to handshape articulation in monomorphemic lexical signs (these constraints are summarized in Section 2.3.2). The HSBN also represents the properties of handshape sequences and the patterns of handshape variation produced in lexical signs to aid in the task of start/end handshape inference given sign language video input.

Since the properties represented in the HSBN are not sign-specific but are properties that are general to a large class of signs in a language, we learn a single HSBN model by utilizing handshape annotations for all lexical signs in the dataset. A sufficient number of examples are therefore available for training the HSBN even in datasets with only a small number of examples for each sign.

### 3.1 HMM models for Sign Language Recognition

HMMs are frequently employed to exploit the sequential structure of phonemes in spoken languages [Ljolje and Levinson, 1991, Jelinek, 1997]. In signed languages, however, phonemes corresponding to different articulatory parameters (which include handshape, hand location, palm orientation and movement type, as described in Section 2.2) are articulated simultaneously as well as sequentially [Battison, 2000]. Factored representations are typically employed to avoid a combinatorial blow-up in the number of HMM states to allow for simultaneous articulatory parameters in signs.

The HMM involves hidden variables whose representation needs to be inferred at training time. In datasets where phonetic transcriptions are not available, a mixture model based representation for articulatory sub-units in signs is learnt from training sequences using either a clustering or an Expectation Maximization (EM) method. Approaches for learning articulatory sub-units are frequently formulated in a generative learning framework, these approaches are described in Section 3.1. A few recent learning approaches for articulatory sub-units have also been formulated with an objective towards retaining discriminative properties between different signs, these are described in Section 3.1. The HSBN representation involves hidden variables to account for the phenomena of handshape variation. The HSBN learning algorithm utilizes handshape label annotations provided by linguists for  $\approx 7,000$  signs contained in the training set. The signs have also been carefully delineated into different lexical items. Given this richer dataset, the HSBN learning algorithm is more strongly constrained than in a fully unsupervised (i.e., without handshape labels) learning formulation. The properties inferred for the latent states in the HSBN are

therefore more easily interpretable (and also verifiable). Furthermore, the learnt HSBN model shows promising improvement in performance over a baseline algorithm for signer-independent handshape recognition. We conclude this section with a few examples of HMM based approaches developed for different SLR applications.

- **The Parallel-HMM formulation**

Vogler and Metaxas [Vogler and Metaxas, 2001] propose the ‘parallel-HMM’ approach assuming independent sequential processes for hand location and movement employing 3D tracks for arms and hands obtained using multiple cameras and physical sensors mounted on the body. The authors define a phonetic representation to describe hand location and hand movement. Phonetic transcriptions for signs in the vocabulary allowing for epenthesis movements (movements occurring at boundaries between two signs in continuous signing) are used to define HMM networks for each channel. The authors show good recognition results for continuous signing sequences constructed from a vocabulary of 22 signs. The authors extend the above approach with handshape information represented by finger angles obtained using a data-glove in [Vogler and Metaxas, 2004].

Von Agris et al. [von Agris et al., 2007] have adopted the parallel-HMM formulation with features for the position and size of the hand along with the spread of fingers. They demonstrate excellent isolated and continuous sign recognition rates for signer-specific recognition on a vocabulary of approximately 230 signs. The signer-independent recognition rates are, however, substantially lower. To improve signer-independent recognition accuracy, the authors propose an approach for model adaptation. A set of annotated utterances produced by the test-signer are used to compute a linear transformation of model parameters estimated from examples in a training set. Handshape inference performance using the HSBN in this thesis is evaluated for the signer-independent application wherein the signs in the test set were produced by a signer whose signs were not included in the training set.

- **Unsupervised learning of articulatory sub-units in parallel-HMMs**

Von Agris et al. [von Agris et al., 2007] propose a sub-unit based representation for signs. Since a lexicon annotated with linguistic attributes was not available when constructing the sub-unit models, the authors propose a *data-driven* decomposition of signs into sub-units. Each sign in the vocabulary is first divided sequentially into segments based on an analysis of the time-series for different articulatory features (these segments have no linguistic interpretation). A subsequent algorithm determines similarities between the identified segments. Similar segments are pooled and are labeled as a single sub-unit. Each sign can thus be described as a sequence of sub-units. The authors report good signer-specific isolated and continuous sign recognition performance employing the sub-unit based representation (the signer wears colored gloves to facilitate hand tracking for the video sequences used in these experiments).

In order to automatically segment signs into sub-units, Han et al. [Han et al., 2009] develop an algorithm based on detecting hand motion discontinuities (i.e., discontinuity in motion speed and trajectory) and using these detections as boundaries for sub-units in signs. They also propose a temporal clustering algorithm using DTW in order to merge similar segments.

- **Discriminative learning of articulatory sub-units in parallel-HMMs**

Simple clustering based approaches for sub-unit construction suffer from the problem of over-fitting to the training set. One approach to reduce the over-fitting problem is to employ a discriminative approach during sub-unit learning. The sought after objective for learning linguistically motivated sub-units can be stated as follows: the learned sub-units should represent articulatory contrast in minimally distinguished pairs of signs. For example, signs within each pair displayed in Figure 2.1 are contrasted in one specific articulation parameter. However, with the small sizes of collected and processed vocabularies available for sign language it is infeasible to exhibit minimally contrasted pairs for each possible configuration for each of the different articulating

parameters. While distinct signs in a lexicon do differ in certain articulatory properties they often share several other articulatory properties. To circumvent this problem, Yin et al. [Yin et al., 2009, Yin, 2010] develop algorithms to implicitly learn articulatory contrast in signs. They propose the segmental-boosting HMM algorithm to construct a transformation of the input feature space to a new feature space in order that observation likelihoods for the HMM trained in the transformed feature space is better able to discriminate among the different hidden states. A second algorithm reduces the number of hidden states by recursively combining states that are most likely to be confused. The authors demonstrate that their algorithm recovers different configurations of feature weights for certain contrasted pairs of signs. Their evaluation is however limited to signs obtained from a single signer.

More recently, Pitsikalis [Pitsikalis et al., 2011] extend the data-driven sub-unit construction approach to include phonetic transcription.

- **Other applications of HMMs for SLR**

Starner et al. [Starner et al., 1998] designed a real-time continuous sign language recognition system based on HMMs for a 40 word lexicon. The features computed from skin region based tracking of the two hands include: each hand’s x and y position, change in x and y between frames, area (in pixels), angle of axis of least inertia (found by the first eigenvector of the blob), length of this eigenvector, and eccentricity of bounding ellipse. Promising results were demonstrated for a signer-dependent recognition task.

A Markov model utilizing multiple articulation parameters was also proposed in [Bowden et al., 2004], however only a small number of handshape classes (6) were considered. A HMM was proposed for fingerspelled word recognition in [Liwicki and Everingham, 2009] using a lexicon consisting of proper nouns (names of people). Legal state transitions in the model correspond to letter sequences for words in the lexicon. In this study, we model linguistic constraints on handshape transitions in lexical signs

(handshape transitions for signs in this class follow certain general rules) and further incorporate variations in handshape across different signers.

### 3.2 Tracking hand articulations in general hand gestures

Many approaches have been proposed to explicitly track finger articulations in a video sequence [Pavlovic et al., 1997, Erol et al., 2007]. However, these approaches impose strong constraints on hand articulation: hands are typically assumed to have little global motion, to occupy a large portion of the video frame, to not overlap with the face or the other hand, and/or to be viewed from certain canonical orientations (the palm of the hand is oriented parallel or perpendicular to the camera). The speed of hand articulation is also assumed to be small. A 2D graphical model and a piecewise planar model for finger articulation are proposed in [Wu et al., 2005, Wang et al., 2008]. Approaches that use a 3D computer graphics hand model [Lu et al., 2003, Tomasi et al., 2003, Sudderth et al., 2004, Chang et al., 2005, Bray et al., 2007, de La Gorce et al., 2008] need good initialization and sufficiently well-resolved hand images in addition to the orientation constraints.

More recently, Oikonomidis et al. [Oikonomidis et al., 2012] have developed a 26 Degree of Freedom (DoF) kinematic model for the hand. They formulate an optimization algorithm to jointly estimate the parameters of the two interacting hands whilst accounting for occlusion relationships and the geometric interactions between the two hands. The authors demonstrate very promising results using RGB+depth input for articulated hand tracking in challenging situations where the two hands are strongly interacting. The range hand poses in their input sequence is, however, limited (the hand configurations correspond approximately to the {5, crvd-5 and A} shapes). In previous work, [Oikonomidis et al., 2011], the authors demonstrate the ability to track a much broader range of hand articulations in a constrained video capture environment using eight calibrated high-resolution cameras. Their method also requires a specification for the 3D kinematic model’s geometry parameters and the joint angle parameters in the first frame for each of the tracking sequences. The input initial configuration aids the optimization algorithm in searching for the optimal



hand pose parameters in subsequent frames.

We offer the following observations as evidence suggesting that handshape classification/inference in signs can in many ways be a *simpler* problem to address than full-DoF articulated hand tracking in unconstrained human gestures. The parameters for different articulations in signs (for example, the configuration, orientation and location as well as the movement trajectory of the hand) are articulated in precise and predictable ways. Some properties that pertain to the articulation of hand configurations in signs are highlighted here. In the class of monomorphemic lexical signs, hand pose parameters are maximally informative at the start and end points of the sign (the speed of articulation between the start and end points of signs is very rapid, however). A certain set of handshapes from among the set of all possible hand configurations are analyzed by linguists as being salient for the purposes of conveying distinctions among different signs. Between 80 and 140 handshapes have for example been employed by linguists for the purposes of analyzing hand configurations in ASL [Whitworth, 2011]. There are constraints that govern how different handshapes can be combined in signs – only a certain number of end handshapes are possible for each start handshape on a given hand; in two-handed signs the non-dominant hand is constrained to either adopt the same start/end handshapes as the dominant hand or to adopt handshapes from a small set of unmarked handshapes. In addition, there are dependencies among the different articulatory parameters such as the handshape, hand location and orientation – only certain handshapes are observed at specific hand locations (wherein the hand locations can be determined with respect to the face, the torso or the other hand), not every handshape configuration is observed for each hand orientation. Van der Kooij [Van der Kooij, 2002] presents both theoretical and empirical analyses of many of these constraints. Our aim in this thesis is to formulate the HSBN representation in order to model the properties and constraints that pertain to the articulation of start/end handshapes in monomorphemic lexical signs.

An observation likelihood distribution is required during handshape inference in order to compute the probabilities of different handshape classes to be associated with hand images

observed in an input video sequence. Recent computer vision approaches have focused their attention on utilizing 3D models along with multiple camera and/or depth input sources to aid with computing observation likelihoods for hand articulation. In SLR applications, however, we are often constrained to a single camera view in order that end-users who possess a webcam can utilize the proposed system. On the database side, an extensive amount of sign language video data has been painstakingly collected and annotated (e.g., [Johnston, 2012, Schembri, 2012, Hanke et al., 2012, Crasborn et al., 2012, Neidle, 2012]) prior to the advent of depth sensors. It is therefore essential that algorithms for SLR be able to utilize the available annotated video data. In this work, the images of start and end handshapes extracted from a large dataset of lexical signs serves as a set of representatives for handshape appearance. In order to account for differences in anthropometry and small variations in handshape articulation among different signers, we employ a non-rigid image alignment algorithm to match the image of a query handshape with handshape images in the database. In future work, we envision that a 3D model based observation likelihood model can be used to augment the image appearance matching based likelihood scores employed in our current implementation.

### 3.3 Handshape recognition in sign language

Active Appearance Model (AAM) based approaches are proposed for general hand pose estimation by Heap and Hogg [Heap and Hogg, 1996], and, for recognizing handshapes in sign language from static images in Fillbrandt et al. [Fillbrandt et al., 2003]. AAM approaches use PCA to capture shape and appearance variations. The learnt modes of variation, however, are tuned to the exemplars in the training set.

Athitsos et al. [Athitsos et al., 2008a] propose a fast nearest neighbor method to retrieve images from a large dictionary of ASL handshapes with similar configurations to a query hand image. The database is composed of renderings from a 3D graphics model for the human hand. The synthetic nature of these images does not yield a robust similarity score to real hand images. Fujimura et al. [Fujimura and Liu, 2006] propose a method

for recognizing hand configurations from depth images, however, no empirical evaluation of their approach is given.

Ding and Martinez [Ding and Martinez, 2009] construct a tree structure to represent landmark locations (fiducials) on the hand. The chosen fiducials correspond to the knuckles (finger joints with the palm), joints within each finger, finger tips and the wrist (palm joint with the lower arm). A fixed handshape is assumed for the duration of a sign. Given 2D coordinates for the visible fiducials in a sequence of frames, the authors propose a linear SVD based reconstruction algorithm allowing for missing data due to self-occlusions to recover the 3D coordinates of the fiducials and global transformations of the hand in each frame. The fiducial co-ordinates in 2D and their visibility attribute are collected interactively with human input: these are initialized in the first frame and predictions in subsequent frames using the reconstruction method are manually verified. Handshape recognition is performed by comparing two 3D hand configurations. The authors report 100% handshape recognition rate using a set of 19 handshapes from 10 subjects producing 38 signs (the training and test sets are from different signers). The results demonstrate a clear benefit in using 3D reconstruction, however, the constraints imposed (handshape is fixed through the sign), the inputs assumed (2D fiducials are given for each frame) and evaluation with signs from participants who are not native users of sign language limits the general applicability of the proposed approach. In our approach, only the hand location bounding boxes at the start and end frames of the sign are assumed as input from the user. We work with a significantly larger collection of signs ( $\approx 3,000$ ) and handshapes (85). Furthermore, signs in our dataset are produced by native ASL users.

Roussos et al. [Roussos et al., 2010] propose an unsupervised clustering formulation in order to extract handshape sub-units (or handshape clusters) from a training set of handshape images. Similar to the AAM approach, a PCA representation is used to model appearance variations for each handshape sub-unit. To impart a degree of robustness to global transformations, the clustering formulation incorporates an affine alignment between a training image and its reconstruction (the reconstructed handshape is computed using the PCA

representation for the handshape sub-unit associated with the training handshape image). Handshapes employed in signing, however, share very similar configurations (Figure 2.3) and are also seen with several different orientations in the training set. These aspects exacerbate the process of clustering training examples into different handshape classes. We circumvent this problem by using a nearest neighbor search to retrieve candidate matches from a database of labeled handshape instances. We develop a non-rigid image alignment method for computing similarity scores between pairs of handshape images and demonstrate its benefits over affine alignment in accommodating inter-signer variations for handshape retrieval.

### 3.4 Appearance features for handshape verification

Image descriptors for handshape appearance are used along with hand location and movement trajectory based features in a sign spotting framework by [Dreuw and Ney., 2008, Alon et al., 2009, Yang et al., 2010]. Farhadi et al. [Farhadi et al., 2007] propose a transfer learning approach, where sign models learnt in a training domain are transferred to a test domain utilizing a subset of labelled signs in the test domain that overlap with those of the training domain (for instance, sign models learnt from one viewpoint can be transferred to a different viewpoint). These approaches do not explicitly distinguish between different handshapes and as a result do not leverage linguistic constraints on handshape transitions.

Buehler et al. [Buehler et al., 2009] describe an approach to automatically extract a video template corresponding to a specified sign gloss (e.g., ‘GOLF’) from TV broadcast continuous signing video with weakly aligned English subtitles. A similarity score for a pair of windowed video sequences is defined based on image features for shape, orientation and location of the hands. This framework, however, treats the sign recognition problem as an instance of a general temporal sequence matching problem and does not exploit phonological constraints on signing parameters. Inter-signer variations are not addressed and the image alignment between hand image pairs is restricted to 2D rotations.

### 3.5 Summary

In summary, while there has been work that has looked at handshape articulation in sign language, none has modeled the linguistic constraints that govern the start and end handshape articulations in lexical signs. We formulate a data-driven probabilistic model, the HSBN, for start/end handshape inference in monomorphemic lexical signs. The HSBN utilizes a layer of hidden variables in order to represent the properties of sign- and signer-independent handshape variation. The properties of hidden variables and the probability distributions that relate the values adopted by these variables are estimated using a variational Bayes learning approach utilizing a training set of monomorphemic lexical signs annotated with start/end handshape labels. A grouping of signs into distinct lexical items is also assumed. Since only a single HSBN model needs to be trained for the entire class of monomorphemic lexical signs, the learning algorithm is able to utilize examples of all signs in this class which in-turn substantially improves the robustness of the estimated HSBN parameters. As a result, the HSBN shows promising improvement in handshape inference performance on a person-independent handshape recognition task. The training set used in our experiment contains 2,636 distinct signs, the corresponding number of productions from 5 signers is 6,958. The test set used in our experiments contains 577 signs produced by a single signer. In query signs from the `two-handed:same handshapes` articulatory class, the HSBN yields further improvement in the handshape recognition accuracy by leveraging bilateral symmetry properties in the handshapes articulated on the two hands.

Task	Hand-shape	References	Approach	Generality of approach
<b>Hand pose tracking for general hand gestures</b>	3D model based	[Stenger et al., 2001, Lu et al., 2003, Tomasi et al., 2003, Sudderth et al., 2004, Chang et al., 2005, Bray et al., 2007, de La Gorce et al., 2008, Oikonomidis et al., 2012]	Track finger articulations using a 3D kinematic model for the hand skeleton	Models self-occlusions; assumes constrained hand orientation, high resolution hand images, good initialization and person specific hand model
	2D model based	[Wang et al., 2008, Wu et al., 2005]	Finger articulation via graphical and piecewise planar models	Initialization not needed; assumes palm orientation parallel to the camera
<b>Sign language handshape recognition from static images</b>	Active Appearance Models	[Fillbrandt et al., 2003]	Uses PCA to capture shape and appearance variations	Fast computation; learnt modes for shape and appearance variation are tuned to the training set
	Nearest neighbor	[Athitsos, 2006, Athitsos et al., 2008a]	Handshapes rendered using 3D model, BoostMap for fast NN	Computationally fast and person-independent; difficult to match synthetic images to real hand images
<b>Handshape classification for sign recognition</b>	Clustering based	[Ong and Bowden, 2004, Bowden et al., 2004]	Markov chain used to model hand shape, movt. and loc. aspects	Good performance on 40 signs; requires colored gloves, inter-person variations are not addressed
<b>Handshape verification</b>	2D appearance features	[Alon et al., 2009, Yang et al., 2010]	Appearance used along with hand trajectory in DTW framework	Used for sign recognition, sign spotting and retrieval; inter-person variations are not handled
<b>Learn video templates for signs from continuous signing video with subtitles</b>	2D appearance	[Buehler et al., 2009, Buehler et al., 2008]	Pictorial structures model for upper body tracking, hand shape, movt., loc. for spatio-temporal similarity	Extracts video templates using multiple instance learning from broadcast video with little supervision; between signer variations not addressed image alignment restricted to 2D-rigid
<b>Phonetic alphabet to describe hand loc. and hand movt.</b>	Finger angles from data-glove	[Vogler and Metaxas, 2001, Vogler and Metaxas, 2004]	Phonetic transcriptions for loc. and movt. aspects, parallel HMM (networks for each channel)	Good recognition results for continuous signing; uses multiple cameras and physical sensors mounted on the body for tracking in 3D

**movt.:** hand movement trajectory, **loc.:** hand location with respect to torso.

**Table 3.1:** A review of approaches for handshape recognition (in sign language) and handpose tracking (general hand gestures).

## Chapter 4

# ASL Lexicon Video Dataset (ASLLVD)

### 4.1 Objectives and requirements for the lexicon dataset

The data collection for this research was carried out with a view towards developing a computer vision system for lexical lookup in sign language video datasets. A ‘lexicon dataset’ containing isolated signs was collected and annotated with linguistic attributes to enable the implementation of a query-by-sign system for sign lookup in an ASL dictionary.

In this chapter we describe the aspects of the lexicon dataset as they pertain to the HSBN formulation for handshape inference in monomorphemic lexical signs. We envision that the lexicon dataset (in concert with other datasets under development for sign language) would facilitate further research into person-independent large vocabulary SLR. Further details about the lexicon dataset are presented in [Neidle et al., 2012b, Athitsos et al., 2008b] and in this webpage [Neidle et al., 2012a].

A large number of distinct signs is needed to provide a representative sample set of lexical items contained in the vocabulary. Productions from multiple native signers are also needed to train recognition methods that are able to accommodate variations in articulation. In this research, the lexicon dataset serves as the primary resource for articulatory patterns in ASL. It is hence essential that videos in the dataset be collected in a controlled environment to facilitate linguistic annotations and the development of reliable and accurate computer models for sign language recognition and retrieval.

Linguistic distinctions between different signs are often attributable to subtle differences in their articulation. To enable computer models to be trained to make such critical distinctions, annotations of several linguistic properties are essential. Some of these required annotations are listed in Section 4.2.4. To the best of our knowledge, previous datasets do

not include all these attributes.

## 4.2 Data collection and annotation

In this section, we present an overview of the methods employed to prepare the lexicon dataset.

The dataset was constructed through collaboration with linguists (under the direction of Carol Neidle), who were responsible for identification of the ASL signers who participated in the project, elicitation of the data, painstaking linguistic annotation and verifications of the annotations for  $\approx 10,000$  signs contained in many hours of sign language video<sup>1</sup>. Several challenges related to ensuring consistency in the annotation of linguistic attributes also needed to be addressed as the annotations were being prepared.

### 4.2.1 Native ASL signers provide signs for the dataset

Six native ASL signers (two men {M1, M2} and four women {F1, F2, F3, F4}) contributed signs for the lexicon dataset. The signers come from a diverse range of geographic and linguistic backgrounds: F1 grew up in Fremont, CA; F2 grew up in Rochester, NY; F3 went to the Minnesota School for the Deaf; F4 and M1 grew up in Boston/Newton, MA; and, M2 went to the Maryland School for the Deaf. The signers range in age from 19 – 40 years.

### 4.2.2 Video capture setup

Videos were captured in a photographic studio with uniform background and controlled illumination. Four time-synchronized color cameras were employed for video capture. Three of these cameras were of standard resolution ( $640 \times 480 @ 60\text{hz}$ ) while the fourth camera was

---

<sup>1</sup>The author wishes to acknowledge the contributions of many people who were involved in developing the ASLLVD. The efforts of our SignStream developer, Iryna Zhuravlova, were instrumental in enabling the preparation of annotations with the precision necessary for the research conducted here. A partial list of native ASL signers and students who participated in the annotation and verification efforts includes: Rachel Benedict, Naomi Berlove, Elizabeth Cassidy, Lana Cook, Alix Kraminitz, Jaimee DiMarco, Joan Nash, Indya Oliver, Caelen Pacelli, Braden Painter, Chrisann Papera, Tyler Richard, Donna Riggle, Tory Sampson, Dana Schlang, Jessica Scott, Jon Suen and Amelia Wisniewski-Barker. Some of the native ASL signers also helped in resolving several difficult questions related to sign language use that arose during data capture and in preparing annotations for the collected data. Computer science contributors include, Vassilis Athitsos, Tianxiong Jiang, Stan Sclaroff, Alexandra Stefan, Gary Wong and Quan Yuan.



of high-resolution ( $1600 \times 1200$  @ 30hz). The camera viewpoints chosen for video capture include: a standard and high-resolution front view of the signer, a side view of the signer and a close-up of the face. To facilitate computer vision based sign language recognition the dataset also includes: video sequences in uncompressed-raw format, camera geometry calibration sequences, and software for skin region extraction.

### 4.2.3 Elicitation methods

Video prompts for the  $\approx 3,000$  signs contained in the Gallaudet Dictionary of American Sign Language [Valli, 2005] were presented to the signers sequentially (in addition to these signs about 500 signs that were not in the dictionary were also elicited). The signs were collected in citation form: the signer is asked to start from a rest position, perform the sign and then return to a rest position. The signers were asked to produce the displayed signs as they naturally would (or not, if they do not use that sign). A total of 9,776 productions were collected from six signers.

The signers did not always produce the same sign as in the prompt, they instead sometimes produced:

- a totally different but synonymous sign;
- a lexical variant of the same sign;
- essentially the same sign but differing in subtle ways with respect to the articulation.

Linguistic annotations of signs in the dataset were thus crucial for the appropriate classification of these productions.

### 4.2.4 Annotation methods

Signs were linguistically annotated using SignStream<sup>®</sup>3. The following annotations are included in general for all signs:

- Timecodes to denote the starting and ending frames of each sign in the video,

- Gloss labels; these are English text that uniquely identifies each sign with a specific item in the sign language vocabulary,
- Articulatory classifications following the typology in Figure 2-4, such as: one-handed/two-handed, same/different handshapes on the two hands in two-handed signs, and, same/different handshapes at start and end positions on each hand,
- Labels for start/end handshapes on one or both hands using the inventory of handshapes displayed in Figure 2-3,
- Morphological classification of sign type (lexical / fingerspelled / loan / classifier constructions) since, as discussed in Section 2.2.3, the compositional principles are different for different classes of signs.

For compound signs, the ASLLVD includes annotations as listed above for each morpheme.

The SignStream interface allows for very detailed and precise annotations to be prepared for signs in the video sequences. However, for the development of sign recognition methods there is also a need to ensure that the annotated linguistic attributes are consistent across different signs in the dataset. The Lexicon Viewer and Verification Tool (LVVT) was therefore developed by the author to aid linguists in viewing, comparing, verifying, and modifying SignStream annotations. The LVVT is designed to assist the annotator in the daunting task of ensuring consistency in the labeling of glosses and articulatory attributes across several thousand productions. The LVVT aids the annotator in the the task of verifying the following attributes for signs: gloss labels, start/end handshapes, start/end timecodes in the video, and the morphological and articulatory classifications of signs. The LVVT facilitates the grouping of signs in order to delineate the different types of variation attested in signs. The ASLLVD includes numeric ID labels to uniquely identify different variants of a sign which further aids in the task of training computer models for recognition. The LVVT allows the verification of the above attributes for morphemes contained within compound signs. Several passes of verifications were made by student annotators, native signers and linguistic experts through the  $\approx 10,000$  productions using

the LVVT to correct a substantial number of linguistic annotations initially prepared by students using SignStream. A more detailed description of features offered by the LVVT is presented in [Neidle et al., 2012b].

### 4.3 The ASLLVD corpus

Statistics for the dataset are summarized in Table 4.1 and are discussed in more detail below. The stimuli employed during elicitation included a total of 2,759 signs. Some stimuli resulted in the production of more than one variant and thus the actual number of signs (including variants) collected is larger: 3,457. With six signers providing the data, a total of 9,776 productions of sign-variants are currently available in the dataset. Signs have been classified into one-handed/two-handed monomorphemic lexical signs, compound constructions, number signs, loan signs, classifier constructions, and fingerspelled signs. Monomorphemic lexical signs are the main focus of this thesis. A total of 2,289 monomorphemic lexical signs are present in the ASLLVD. Including all variants this number is 2,923 and the corresponding number of productions obtained from all six signers combined is 8,562.

Signs in the dataset have been classified by linguists with careful consideration of different articulatory properties (e.g., Figure 4-1). Annotation of these distinctions is an essential step towards training computer models that are capable of making the same distinctions. A unique gloss label is associated with each distinct sign-variant. For 73% of sign-variants in the class of monomorphemic lexical signs, productions from more than one signer are available. Articulatory variations are often observed among productions that correspond to a given gloss label. The lexicon dataset can therefore serve as a valuable resource to further the development of SLR methods that are able to accommodate articulatory variation.

We use ACCIDENT and APPOINTMENT as examples to describe the organization of signs in the lexicon dataset. These signs were chosen because a wide range of interesting handshape variation was attested among their different productions in the dataset. The annotations prepared by linguists for the productions of these two signs are displayed

Classification of signs		Number of signs	Number of signs (variants)	# sign-variants with {1, 2, ..., 6} consultants	# tokens (examples) per sign-variant {1, 2, ..., 6, >6}	Number of sign tokens		
Monomorphemic lexical signs	Two-handed	1557	1960	×1	537	503	×1	5687
				×2	679	587	×2	
				×3	273	267	×3	
				×4	341	295	×4	
				×5	55	100	×5	
				×6	75	100	×6	
			108	>6				
	One-handed	824	968	×1	245	240	×1	2875
				×2	312	266	×2	
				×3	136	138	×3	
				×4	189	164	×4	
				×5	31	47	×5	
				×6	55	58	×6	
			55	>6				
	Subtotal of above	2289 <sup>(a)</sup>	2923 <sup>(b)</sup>	×1	<b>777</b>	<b>738</b>	×1	8562
				×2	<b>990</b>	<b>852</b>	×2	
				×3	<b>410</b>	<b>405</b>	×3	
				×4	<b>529</b>	<b>458</b>	×4	
×5				<b>87</b>	<b>149</b>	×5		
×6				<b>130</b>	<b>157</b>	×6		
		<b>164</b>	>6					
Compound signs		291	346	×1	151	139	×1	746
			×2	109	107	×2		
			×3	46	46	×3		
			×4	28	31	×4		
			×5	5	11	×5		
			×6	7	10	×6		
					2	>6		
Numbers		78	103					260
Loan signs		46	52					136
Classifier constructions		29	32					41
Fingerspelled signs		21	21					25
<b>ALL</b>		<b>2759<sup>(c)</sup></b>	<b>3457<sup>(d)</sup></b>	--	--	--	--	<b>9776</b>

Counts in cells <sup>(a,b,c,d)</sup> are less than the totals of counts in their parent cells because of the following reasons: **(a)** many signs often contain both one- and two-handed variants; adding the parent cells will count these signs twice, **(b)** minor annotation inconsistencies occur where one- and two-hand tokens have been incorrectly placed in the same variant collection and this contributes some extra counts, **(c)** & **(d)** are very similar to **(a)** & **(b)** in that there are a few instances of conflation across different classes.

**Table 4.1:** Statistics for signs contained in the ASLLVD corpus.

in Table 4.2. Figures 4.2 and 4.3 show the start/end frames for these sign productions. We summarize the variations found for productions associated with these two example signs in Table 4.3. The number of signers from whom each of the two signs were elicited are in the first row. Variations attested among the productions of each sign are classified as follows:

- **Lexical variants**

Three linguistically distinct variants – considered here to be three distinct lexical items – are attested in both cases. These variants are differentiated from each other in certain specific handshapes that are articulated on either one or both hands at either the start or end positions of the sign. For example, although a sign corresponding roughly to the meaning of the English word, "appointment" can be produced with either a start handshape of A or 5, these handshapes generally cannot be used interchangeably in other signs without changing meaning. Therefore variations classified here as lexical variations reflect possibilities for specific lexical items rather than general phonological processes in the language. The differences among these variants are summarized in the fourth row of Table 4.3. Each lexical item is annotated with a unique gloss label. The number of examples associated with each lexical item is given in the fifth row of the same table.

- **Variations produced as a result of general phonological processes**

The multiple examples available for many of the lexical variants allow us to extract the patterns of handshape variation that are attributable to general language phenomena (i.e., phonological variation). The patterns of variation that are the focus of this research are those produced without being influenced by the phonological environment within which the handshape appears. This is because the productions of signs used in this research were all produced in isolation (i.e., in citation form). Sign-independent handshape variations attested among the productions of ACCIDENT and APPOINTMENT in the ASLLVD are described in the last row of Table 4.3. Handshape

variations in other example signs from the dataset are illustrated in Figure 4.4. The productions of each sign obtained from different signers are depicted in the table using annotations for the start/end handshapes. The handshape labels associated with different examples of a given sign are grouped by utilizing the linguistic properties associated with the articulatory class (*one-handed / two-handed : same handshapes / two-handed : different handshapes*) that the sign belongs to. These groupings of handshape labels are outlined in the chart using boxes drawn with distinct colors. The sign BLOSSOM, for example, shares the same handshape on the dominant and non-dominant hands. The start and end handshapes in this sign are different, however.

Returning to Table 4.1, we now describe the different columns in more detail: There are a total of 2,923 sign-variants (column 3) among monomorphemic lexical signs. For 777 of those sign-variants (column 4), we have examples from only one signer; for 990 of them, we have examples from two signers, etc., and for 130 of those sign-variants, we have examples from all six of our native signers. Since we have more than one example per signer for some sign-variants, the total number of examples per sign-variant may be greater than the total number of signers whose productions of that sign-variant are included in our data set. In fact, for 164 of the sign-variants (column 5), we have more than 6 examples. (For two of the signs, we have as many as 19 productions.)

	Consultant	Main Gloss	Gloss Variant	Dominant hand start HS	Non-dominant hand start HS	Dominant hand end HS	Non-dominant hand end HS	
sign variant 1	F1	ACCIDENT	ACCIDENT	S	S	S	S	
	F2		ACCIDENT	S	S	S	S	
	-----							
	F1		(5)ACCIDENT	5	5	A	A	
sign variant 2	F2		(5)ACCIDENT	5	5	10	10	
	F3		(5)ACCIDENT	5	5	S	S	
	F4		(5)ACCIDENT	5	5	S	S	
	-----							
sign variant 3	M1		(3)ACCIDENT	3	3	S	A	

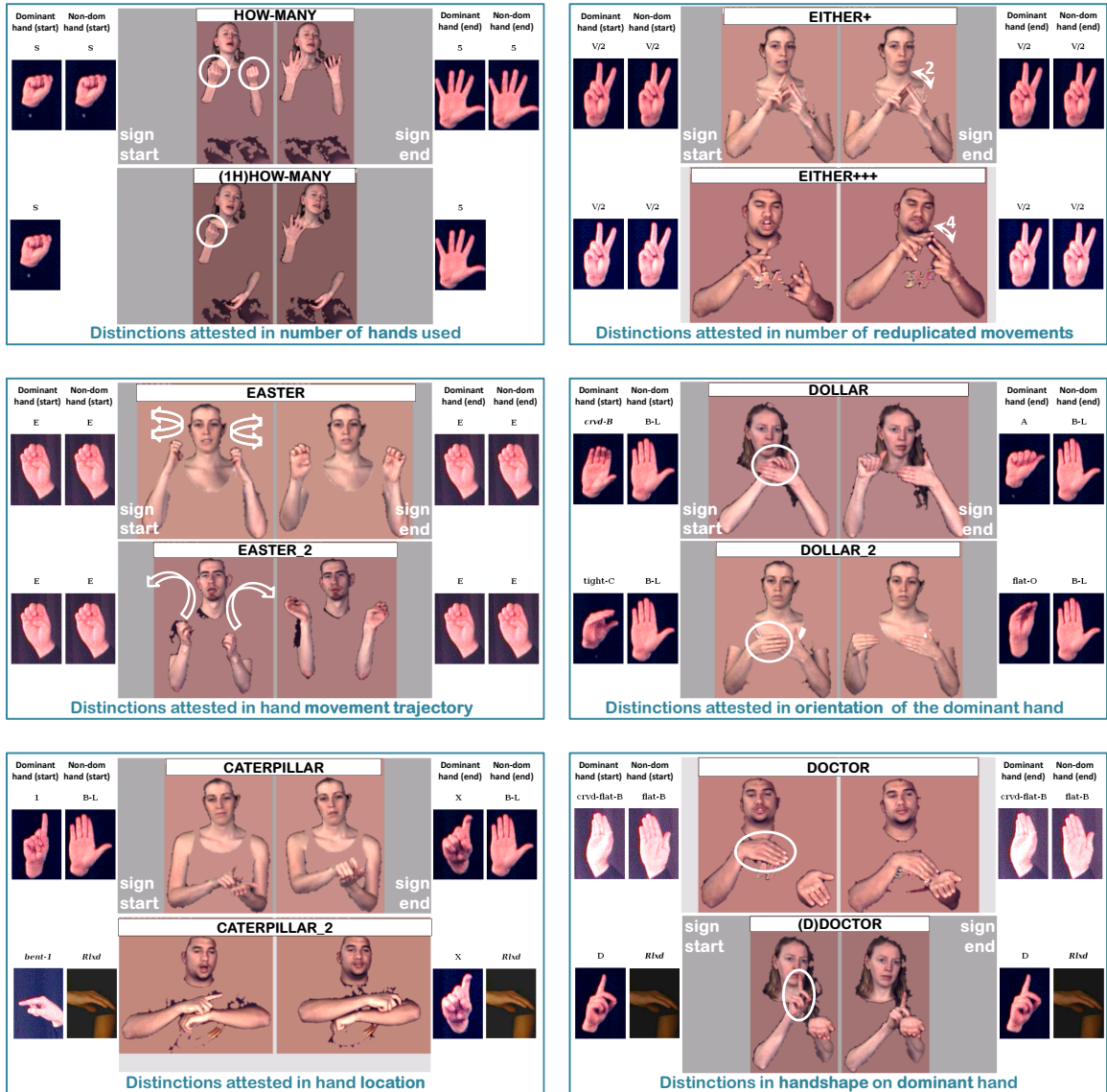
  

	Consultant	Main Gloss	Gloss Variant	Dominant hand start HS	Non-dominant hand start HS	Dominant hand end HS	Non-dominant hand end HS	
sign variant 1	F1	APPOINTMENT	APPOINTMENT	5	5	A	A	
	M1		APPOINTMENT	5	crvd-flat-B	S	S	
	M1		APPOINTMENT	5	crvd-5	S	S	
	F2		APPOINTMENT	5	crvd-5	S	S	
	F2		APPOINTMENT	crvd-5	crvd-5	S	S	
	M2		APPOINTMENT	5	5	A	A	
	F3		APPOINTMENT	5	crvd-5	S	S	
	F4		APPOINTMENT	5	5	S	S	
	-----							
	sign variant 2		F1	(A)APPOINTMENT	A	A	A	A
-----								
sign variant 3	F1		(nd-S)APPOINTMENT	crvd-sprd-B	S	S	S	
	M2	(nd-S)APPOINTMENT	crvd-5	S	A	S		

**Table 4.2:** Handshape labels and attested variants among examples of ACCIDENT and APPOINTMENT contained in the lexicon dataset.

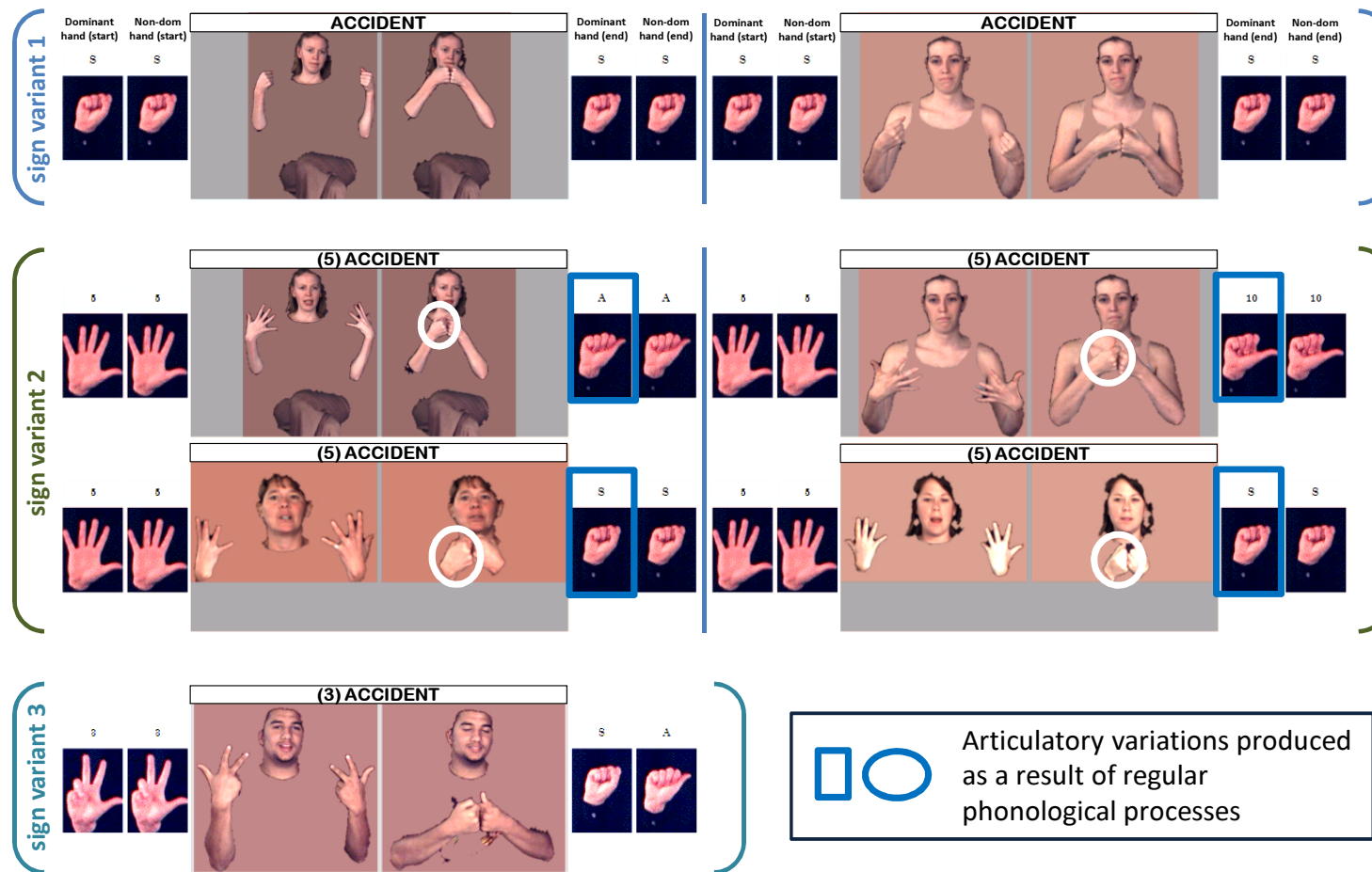
↓ Properties attested in the lexicon dataset	ACCIDENT	APPOINTMENT
Total number of examples	7	11
Number of signers who produced the given sign	5	6
Number of linguistically distinct forms (i.e., <b>sign-variants</b> ) among the sign productions contained in the dataset, and their annotated glosses	3 (a) ACCIDENT (b) (5)ACCIDENT (c) (3)ACCIDENT	3 (a) APPOINTMENT (b) (A)APPOINTMENT (c) (nd-S)APPOINTMENT
Articulatory differences that produce <b>linguistic distinctions</b> in these signs	(b) Differs from (a) in start handshape on both hands (c) Differs from (a) & (b) in start handshape on both hands	(b) Differs from (a) in start handshape on both hands  (c) Differs from (a) & (b) in start handshape on the non-dominant hand
Number of examples for each sign-variant	2, 4, 1	8, 1, 2
Articulatory differences likely to be produced as a result of <b>regular phonological processes</b> (modeling these processes is the focus of this thesis)	(b) {A,10,S} variation in end handshape on both hands	(a) {5,crvd-5,crvd-flat-B} variation in start handshape on the non-dominant hand, {A,S} variation in end handshape on both hands  (c) {crvd-sprd-B,crvd-5} variation in start handshape on dominant hand, {A,S} variation in end handshape on dominant hand

**Table 4.3:** A summary of different variations observed in the lexicon dataset for the signs ACCIDENT and APPOINTMENT based on annotations for different productions of these signs as listed in Table 4.2.



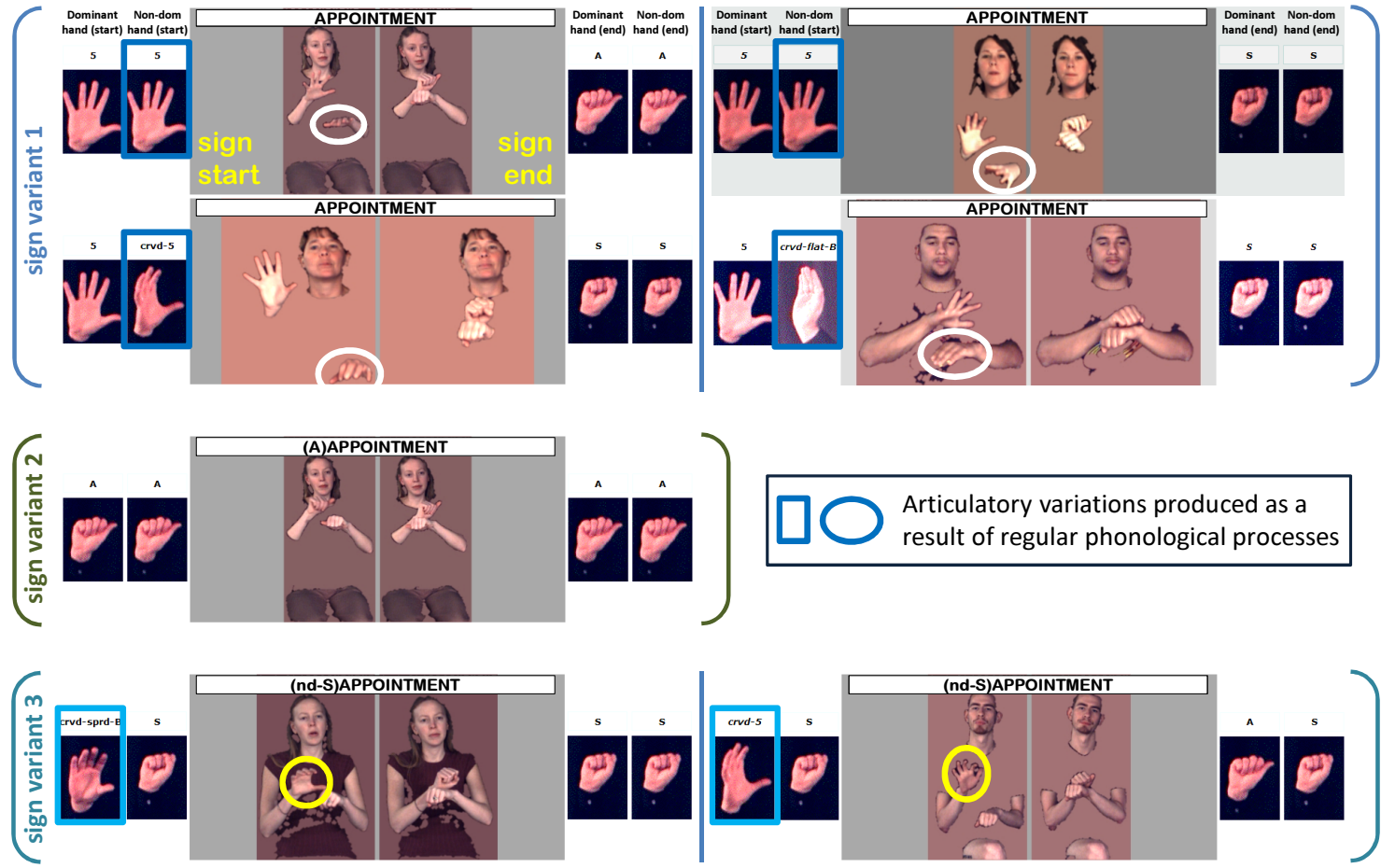
**Figure 4-1:** Annotations in the ASLLVD delineate variations in articulatory features that are linguistically distinctive. A few examples of such distinctions in different articulatory parameters are shown here.





**Figure 4-2:**

Examples of variations attested for the sign ACCIDENT in the ASLLVD. The three lexical variants of this sign are annotated with the gloss labels ACCIDENT, (5)ACCIDENT, (3)ACCIDENT. Phonological variations in the end handshape (e.g., A, S, 10) are seen among the four examples of (5)ACCIDENT.



**Figure 4.3:** Examples of variations attested for the sign APPOINTMENT in the ASLLVD. The three lexical variants of this sign are annotated with the gloss labels APPOINTMENT, (A)APPOINTMENT, (nd-S)APPOINTMENT. Phonological variations in the end handshape of the non-dominant hand are seen among the examples of APPOINTMENT and in the start handshape of the dominant hand among the examples of (nd-S)APPOINTMENT.

### 4.3.1 Limitations in relying on handshape annotations as the ground-truth

One important aspect pertaining to the implementation of the proposed approach for modeling the properties of handshape combinations as well as the properties of handshape variations within the HSBN relates to the fact that the start/end handshapes annotated for signs in the lexicon dataset are assumed to reflect the ground-truth hand configurations articulated within a given video sequence. Preparing handshape annotations is inherently subjective due to the difficulties involved in associating a particular label from among a finite set of handshape classes to hand configurations that are observed as start/end hand images in the input signing video. Since hand configurations observed in signs often do not exactly match one of the predetermined set of handshapes, the annotators had to make a forced choice (the apparent difference in handshapes in some signs may therefore be greater than the actual difference in the hand configurations). Hands in many cases are only partially visible due to both self-occlusions and occlusions produced by the other hand. Differences in handshape annotations can also arise from differences in the start and end frames selected by annotators for multiple productions of a sign. All these factors are additional sources of differences/variations in the sets of handshape labels for a given sign-variant that are employed for training the HSBN model. Therefore, a prior over the model parameters is incorporated during the learning of the HSBN in order to improve the robustness of the estimated parameters.

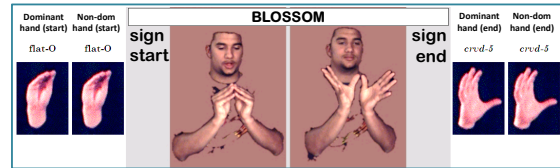
## 4.4 Summary

The lexicon dataset was prepared with a goal of facilitating the development of a query-by-sign lookup system for an ASL dictionary. The lexicon dataset is unique in that it includes extensive annotations painstakingly prepared by linguists for several attributes of signs, with a specific focus on the properties of hand articulations. The annotations that are available for productions of signs contained in the dataset include the start/end video frames, the start/end handshapes, as well as morphological and articulatory classifications of signs. With the goals of distinguishing between variations in articulation that occur in

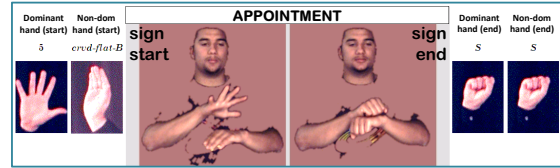
Handshape annotations for the productions of four signs in the ASLLVD

An example production of each sign along with its start/end handshape annotations

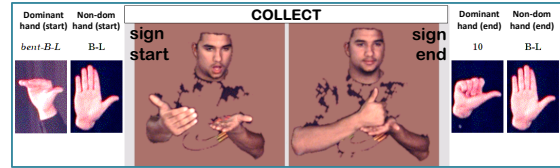
Sign	Signer	Dominant hand START handshape	Non-dominant hand START handshape	Dominant hand END handshape	Non-dominant hand END handshape
BLOSSOM	Li	flat-O	flat-O	5	5
	Ty	flat-O	flat-O	crvd-5	crvd-5
	Na	O	O	crvd-sprd-B	crvd-sprd-B
	Br	flat-O	flat-O	5	5



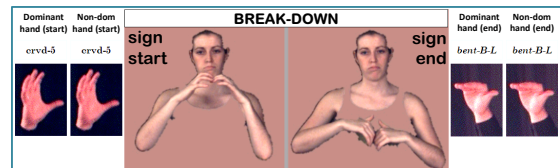
Sign	Signer	Dominant hand START handshape	Non-dominant hand START handshape	Dominant hand END handshape	Non-dominant hand END handshape
APPOINTMENT	Li	5	5	A	A
	Ty	5	crvd-flat-B	S	S
	Na	crvd-5	crvd-5	S	S
	Br	5	5	A	A
	La	5	crvd-5	S	S
	Da	5	5	S	S



Sign	Signer	Dominant hand START handshape	Non-dominant hand START handshape	Dominant hand END handshape	Non-dominant hand END handshape
COLLECT	Li	crvd-B	B-L	10	B-L
	Li	crvd-5	B-L	A	B-L
	Ty	bent-B-L	B-L	10	B-L
	Na	5	B-L	10	B-L
	Na	crvd-5	B-L	10	B-L
	Br	5	B-L	A	B-L



Sign	Signer	Dominant hand START handshape	Non-dominant hand START handshape	Dominant hand END handshape	Non-dominant hand END handshape
BREAK-DOWN	Li	crvd-5	crvd-5	crvd-5	crvd-5
	Li	5-C-L	5-C-L	crvd-5	crvd-5
	Na	crvd-5	crvd-5	bent-B-L	bent-B-L
	Br	crvd-5	crvd-5	crvd-5	crvd-5



**Figure 4-4:** Examples of handshape variation attested in the ASLLVD corpus. The focus here is on patterns of handshape variation that are produced as a result of general language processes. These are handshape variations that are not tightly linked to a specific item in the vocabulary. The start/end handshape labels on the dominant and non-dominant hands annotated by linguists are shown in the left column for examples of selected signs. An example for each sign (dashed outline) is depicted in the right column.

general across the language and those variations that are, for the most part, particular to certain specific items in the vocabulary, the productions of distinct signs have been annotated with a unique gloss (these are text labels in English). Multiple productions of signs, in many instances from different signers, are available for a large fraction of signs in the dataset vocabulary. In total, the lexicon dataset includes 9,776 productions of 3,457 distinct signs.

We envision that the lexicon dataset can serve as a valuable resource for developing data-driven approaches for learning the properties of articulation as well as the patterns of articulatory variation observed in signs. In this research we will utilize the lexicon dataset specifically for the purposes of learning and empirical evaluation of the HSNB formulation for the task of handshape inference in monomorphemic lexical signs.

## Chapter 5

# HandShapes Bayesian Network (HSBN)

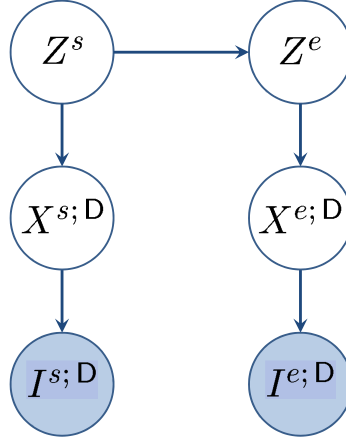
In this chapter, we aim to formulate probabilistic models to represent the properties of start/end handshape combinations in monomorphemic lexical signs. The models are developed with an eye towards facilitating start/end handshape inference given video input of a sign. As summarized in the preceding chapter on ASL linguistics, the three main articulatory classes of monomorphemic lexical signs are:

- (a) **‘two-handed : same handshapes’**: the handshapes articulated on the two hands are the same (or, are very similar),
- (b) **‘two-handed : different handshapes’**: the two-hands exhibit dissimilar handshapes in either or both the start and end points of the sign. The non-dominant hand takes a small subset of possible handshapes and also does not exhibit a change in handshape between the start and end positions, and
- (c) **‘one-handed’**: only the dominant hand is involved in the articulation.

We will propose a HandShape Bayesian Network (HSBN) model for each of these three articulatory classes. An HSBN is a probabilistic generative model that represents the likely combinations of start/end handshapes in monomorphemic lexical signs. We start by formulating the HSBN for the class of one-handed signs, and then extend this model to obtain the HSBNs for two-handed signs. The mathematical notation used in the HSBN formulation is summarized in Table 5.1.

Notation	Description
$I^{s;D}, I^{e;D}$	Images of handshapes for the dominant hand observed in the input video at the start and end of the sign
$I^{s;N}, I^{e;N}$	Images of handshapes for the non-dominant hand observed in the input video in two-handed signs
$\mathcal{X}$	Inventory of handshape labels, which contains 85 handshape distinctions in our implementation
$X^{s;D}, X^{e;D}, X^{s;N}, X^{e;N}$	Handshape labels from the set $\mathcal{X}$ for the observed start/end handshape images $\mathbf{i}^{s;D}, \mathbf{i}^{e;D}, \mathbf{i}^{s;N}, \mathbf{i}^{e;N}$
$Z^s, Z^e$	Variables depicting hidden (unobserved) start/end states
$\mathbf{Z} = (Z^s, Z^e)$	State-space associated with the hidden variables $Z^s, Z^e$ , which are estimated during HSBN learning.

**Table 5.1:** Notations used in the HSBN formulation.



**Figure 5.1:** The HSBN<sup>dominant</sup> graphical model for handshape inference in one-handed signs.

### 5.1 HSBN for one-handed signs

For one-handed signs, the dominant hand alone participates in the articulation. Thus, our model for one-handed signs considers only the start and end handshapes of the signer’s dominant hand. The corresponding HSBN<sup>dominant</sup> model is depicted in Figure 5.1. The model comprises three layers of random variables. The lowest layer represents handshape images observed for the dominant hand at the start and end positions of the sign. The

images of the dominant hand are denoted using the random variables  $I^{s;D}, I^{e;D}$ . The middle layer in the model includes the random variables,  $X^{s;D}, X^{e;D}$ , to depict handshape labels for the start/end handshape images. The inventory of handshapes,  $\mathcal{X}$ , in our implementation contains 85 labels. The top layer of the HSBN model accounts for the hidden variables. The labels for observed handshapes  $X^{s;D}, X^{e;D}$  in the HSBN are obtained as different realizations of certain hidden states,  $Z^s, Z^e$ . Hidden variables are included in the HSBN to model the phenomena of handshape variation produced as a result of general phonological processes. The phenomena of sign-independent phonological variation are described in more detail in Chapters 2 and 4.

The HSBN is formulated for the handshape classification task wherein labels from a pre-defined set of handshapes,  $\mathcal{X}$ , are desired as outputs of the handshape inference algorithm. A convenient modeling choice for the HSBN is to employ a collection of discrete states to represent hidden variables. Probability distributions that involve the hidden variables,  $Z^s, Z^e$ , reduce to multinomial distributions, a property that enables relatively efficient algorithms for HSBN learning and handshape inference. Handshapes in signs are produced as a result of the hands adopting configurations in a continuous parameter space and therefore robustness to gradience in handshape configurations is essential in algorithms for handshape inference. In the proposed HSBN implementation, a degree of robustness to small differences in articulation is incorporated into the observation likelihood function by using an algorithm for non-rigid handshape image alignment. An alternate modeling choice for the hidden variables that utilizes a continuous domain representation (such as a Gaussian mixture model) requires a significantly larger training set size in order to accommodate the wide range of hand orientations attested in signs. Furthermore, several handshapes are either indistinguishable or are very similar in many of their 2D projections. We set aside the investigation of a continuous domain representation for hidden variables as a topic for future work.

Given the assumed representation for hidden variables in the HSBN, the probability distributions in the model and their associated parameters are defined as follows. The



Notation	Description
$\boldsymbol{\pi}_{z^s}$ or $\boldsymbol{\pi}[z^s]$	The prior distribution $P(Z^s = z^s)$ , for the hidden state at the start of a sign
$\mathbf{a}_{z^s, z^e}$ or $\mathbf{a}[z^s, z^e]$	Transition probabilities $P(Z^e = z^e   Z^s = z^s)$ for start/end hidden states
$\mathbf{b}_{z^s}^s(x^s)$ or $\mathbf{b}^s[z^s, x^s]$ , $\mathbf{b}_{z^e}^e(x^e)$ or $\mathbf{b}^e[z^e, x^e]$	The probabilities for observed handshape labels to be obtained as different realizations of hidden states: $P(X^s = x^s   Z^s = z^s)$ , $P(X^e = x^e   Z^e = z^e)$
$\boldsymbol{\lambda}$	The parameters $\{\boldsymbol{\pi}, \mathbf{a}, \mathbf{b}^s, \mathbf{b}^e\}$ for the HSBN model

**Table 5.2:** Parameters for the HSBN formulation.

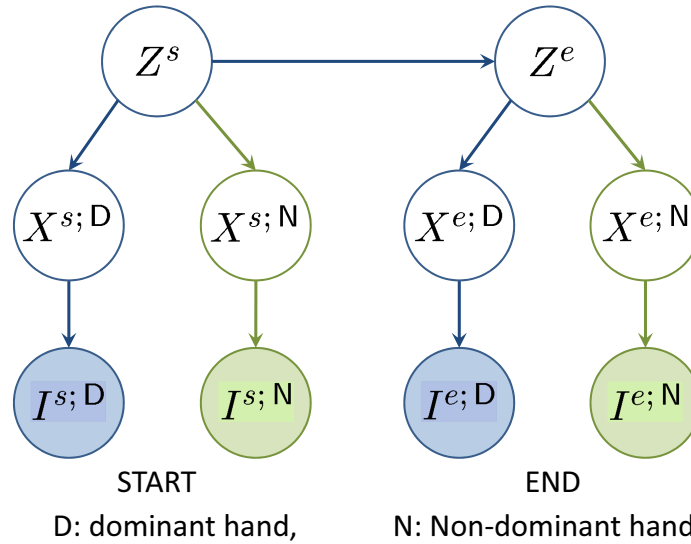
probability distribution over the start latent states are denoted as:  $\boldsymbol{\pi}_{z^s} = P(Z^s = z^s)$ . The start/end transitions in the model are represented as:  $\mathbf{a}_{z^s, z^e} = P(Z^e = z^e | Z^s = z^s)$ . The probability distributions for observed handshape configurations to be produced as different realizations of hidden states are given by  $\mathbf{b}_{z^s}^s(x^{s;D}) = P(X^{s;D} = x^{s;D} | Z^s = z^s)$  and  $\mathbf{b}_{z^e}^e(x^{e;D}) = P(X^{e;D} = x^{e;D} | Z^e = z^e)$ . These parameters taken together are denoted as  $\boldsymbol{\lambda}$  and are summarized in Table 5.2.

The likelihoods of producing the observed start/end handshape appearances in input video given their corresponding handshape configuration labels are depicted as:  $P(I^{s;D} = \mathbf{i}^{s;D} | X^{s;D} = x^{s;D})$  and  $P(I^{e;D} = \mathbf{i}^{e;D} | X^{e;D} = x^{e;D})$ . The expressions for these distributions are derived in a subsequent section on handshape inference.

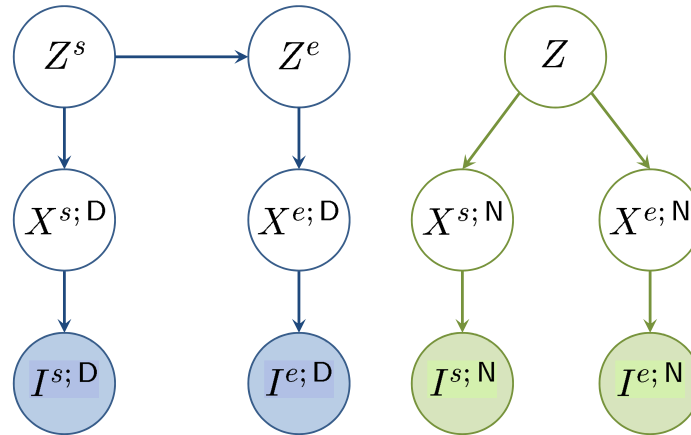
## 5.2 HSBN for two-handed signs

Signs in the class `two-handed:same handshapes` exhibit same or very similar handshapes on the two hands. The pair of random variables  $X^{s;D}, X^{s;N}$  for handshape configurations on the two hands at the start of the sign are therefore modelled as different realizations of the same hidden state  $Z^s$ . Similarly, the handshape pair  $X^{e;D}, X^{e;N}$  observed at the end of the sign are modelled as different realizations of the hidden state  $Z^e$ . The corresponding HSBN<sup>congruent</sup> graphical model is depicted in Figure 5.2.

In the case of `two-handed:different handshapes` signs, the handshapes of the dominant



**Figure 5-2:** The  $\text{HSNB}^{\text{congruent}}$  graphical model formulated for handshape inference in two-handed : same handshapes signs.



**Figure 5-3:** The  $\text{HSNB}^{\text{dominant}}$  and  $\text{HSNB}^{\text{non-dominant}}$  graphical models formulated for handshape inference in two-handed : different handshapes signs.

hand are represented using the same graphical model as in the case of one-handed. The non-dominant hand adopts configurations from among a small set of unmarked handshapes and the non-dominant hand does not exhibit a change in configuration between the start and end positions of the sign. The properties that are unique to handshapes articulated on the non-dominant hand are therefore represented using the  $\text{HSNB}^{\text{non-dominant}}$  graphical model depicted in Figure 5.3. The same hidden state  $Z$  produces the observed start and

end handshapes  $X^{s;N}, X^{e;N}$ .

The handshapes on the non-dominant hand in `two-handed : different handshapes` are dealt with separately in learning the HSBN model. In all the other cases, the hands can adopt a full range of handshapes (Figure 2.3(a)); furthermore, they share the same patterns and constraints with respect to the allowable changes in handshape configuration between the start and end points in the sign. The algorithm for learning the HSBN parameters can therefore utilize examples in the training set that belong to all three classes. At query time, different versions of the HSBN, as determined by the class to which the query sign belongs to, are constructed in order to perform handshape inference.

### 5.3 Handshape inference using the HSBN model

Given the HSBN representations described above, we now formulate the proposed approach for handshape inference for an input video of a sign. We first develop the approach for handshape inference in `one-handed` query signs and then extend this formulation for handshape inference in `two-handed` query signs.

#### 5.3.1 Handshape inference in one-handed signs

Given start/end handshape images  $\mathbf{i}^{s;D}, \mathbf{i}^{e;D}$  for the dominant hand in an input video sequence, we would like to infer the likely start/end handshape labels. The  $\text{HSBN}^{\text{dominant}}$  model yields the posterior probability distribution  $P(X^{s;D} = x^{s;D}, X^{e;D} = x^{e;D} | I^{s;D} = \mathbf{i}^{s;D}, I^{e;D} = \mathbf{i}^{e;D})$  for the start/end handshape labels which can then be used to produce the inferred list start/end handshape pairs. The posterior

distribution, in the HSBN<sup>dominant</sup> graphical model in Figure 5.1, is computed as follows,

$$\begin{aligned} & P(X^{s;\text{D}} = x^{s;\text{D}}, X^{e;\text{D}} = x^{e;\text{D}} \mid I^{s;\text{D}} = \mathbf{i}^{s;\text{D}}, I^{e;\text{D}} = \mathbf{i}^{e;\text{D}}) \\ &= \frac{P(x^{s;\text{D}}, x^{e;\text{D}}, \mathbf{i}^{s;\text{D}}, \mathbf{i}^{e;\text{D}})}{P(\mathbf{i}^{s;\text{D}}, \mathbf{i}^{e;\text{D}})} \end{aligned} \quad (5.1)$$

$$\begin{aligned} &\propto P(x^{s;\text{D}}, x^{e;\text{D}}, \mathbf{i}^{s;\text{D}}, \mathbf{i}^{e;\text{D}}) \\ &= P(\mathbf{i}^{s;\text{D}}, \mathbf{i}^{e;\text{D}} \mid x^{s;\text{D}}, x^{e;\text{D}}) P(x^{s;\text{D}}, x^{e;\text{D}}) \\ &= P(\mathbf{i}^{s;\text{D}} \mid x^{s;\text{D}}) P(\mathbf{i}^{e;\text{D}} \mid x^{e;\text{D}}) P(x^{s;\text{D}}, x^{e;\text{D}}) \\ &= P(\mathbf{i}^{s;\text{D}}) P(\mathbf{i}^{e;\text{D}}) \frac{P(x^{s;\text{D}} \mid \mathbf{i}^{s;\text{D}})}{P(x^{s;\text{D}})} \frac{P(x^{e;\text{D}} \mid \mathbf{i}^{e;\text{D}})}{P(x^{e;\text{D}})} P(x^{s;\text{D}}, x^{e;\text{D}}) \\ &\propto P(x^{s;\text{D}} \mid \mathbf{i}^{s;\text{D}}) P(x^{e;\text{D}} \mid \mathbf{i}^{e;\text{D}}) \frac{P(x^{s;\text{D}}, x^{e;\text{D}})}{P(x^{s;\text{D}}) P(x^{e;\text{D}})}. \end{aligned} \quad (5.2)$$

In order to evaluate the above expression, we need to specify the posterior form for the observation likelihoods,  $P(x^{s;\text{D}} \mid \mathbf{i}^{s;\text{D}})$ , and the prior distribution over handshape label pairs,  $P(x^{s;\text{D}}, x^{e;\text{D}})$ . We discuss one specific implementation for the observation likelihood function in Chapter 8. The expression for the HSBN observation likelihood is formulated in Equation 8.1.

The HSBN<sup>dominant</sup> model yields the following decomposition for the prior distribution over handshape label pairs in terms of the model parameters,  $\boldsymbol{\lambda}$ :

$$\begin{aligned} & P(x^{s;\text{D}}, x^{e;\text{D}}) \\ &= \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} P(Z^s = z^s) P(X^{s;\text{D}} = x^{s;\text{D}} \mid Z^s = z^s) P(Z^e = z^e \mid Z^s = z^s) P(X^{e;\text{D}} = x^{e;\text{D}} \mid Z^e = z^e) \\ &= \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} \boldsymbol{\pi}_{z^s} \mathbf{a}_{z^s, z^e} \mathbf{b}_{z^s}^s(x^{s;\text{D}}) \mathbf{b}_{z^e}^e(x^{e;\text{D}}). \end{aligned} \quad (5.3)$$

The distributions  $P(x^{s;\text{D}})$ ,  $P(x^{e;\text{D}})$  are computed as marginals of  $P(x^{s;\text{D}}, x^{e;\text{D}})$ .

Substituting Equations 8.1 and 5.3 into Equation 5.2 completes the steps for handshape inference in one-handed signs.

### 5.3.2 Handshape inference in two-handed signs

For the task of handshape inference in two-handed signs we require a list of labels for start/end handshapes attested on the dominant and non-dominant hands in a video sequence containing a two-handed sign. The respective start/end handshape images are denoted as  $\mathbf{i}^{s;D}$ ,  $\mathbf{i}^{e;D}$ , and,  $\mathbf{i}^{s;N}$ ,  $\mathbf{i}^{e;N}$ .

In `two-handed:same handshapes` signs, same (or very similar) handshapes are articulated on the two hands. To infer the respective handshape labels we compute the following joint posterior probability distribution:  $P(x^{s;D}, x^{e;D}, x^{s;N}, x^{e;N} | \mathbf{i}^{s;D}, \mathbf{i}^{e;D}, \mathbf{i}^{s;N}, \mathbf{i}^{e;N})$ . Utilizing the `HSBNcongruent` graphical model in Figure 5.2, an expression for the joint posterior distribution is derived as follows. We first obtain the following expression following the sequence of steps as in Equations 5.1 and 5.2,

$$\begin{aligned} & P(x^{s;D}, x^{e;D}, x^{s;N}, x^{e;N} | \mathbf{i}^{s;D}, \mathbf{i}^{e;D}, \mathbf{i}^{s;N}, \mathbf{i}^{e;N}) \\ & \propto P(x^{s;D} | \mathbf{i}^{s;D}) P(x^{e;D} | \mathbf{i}^{e;D}) P(x^{s;N} | \mathbf{i}^{s;N}) P(x^{e;N} | \mathbf{i}^{e;N}) \frac{P(x^{s;D}, x^{e;D}, x^{s;N}, x^{e;N})}{P(x^{s;D}) P(x^{e;D}) P(x^{s;N}) P(x^{e;N})}. \end{aligned} \quad (5.4)$$

The posterior form of the observation likelihoods,  $P(x | \mathbf{i})$ , are computed as in Equation 8.1. The `HSBNcongruent` graphical model yields the following decomposition of the prior probability distributions for the observed handshape labels:

$$P(x^{s;D}, x^{e;D}, x^{s;N}, x^{e;N}) = \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} \boldsymbol{\pi}_{z^s} \mathbf{a}_{z^s, z^e} \mathbf{b}_{z^s}^s(x^{s;D}) \mathbf{b}_{z^s}^s(x^{s;N}) \mathbf{b}_{z^e}^e(x^{e;D}) \mathbf{b}_{z^e}^e(x^{e;N}). \quad (5.5)$$

The distributions  $P(x^{s;D})$ ,  $P(x^{e;D})$ ,  $P(x^{s;N})$ ,  $P(x^{e;N})$  are computed as marginals of the joint prior distribution  $P(x^{s;D}, x^{e;D}, x^{s;N}, x^{e;N})$ .

Substituting the expressions for the joint prior distribution and the posterior form of observation likelihoods into Equation 5.4 yields the desired posterior distributions for handshape label tuples.

In `two-handed:different handshapes` signs, the posterior distributions for handshape labels on the dominant hand are computed using the same sequence of steps as for handshape inference in `one-handed` signs (Section 5.3.1). The posterior distributions for handshapes on

the non-dominant hand are computed as follows.

$$\begin{aligned}
 & P(X^{s;\mathbf{N}} = x^{s;\mathbf{N}}, X^{e;\mathbf{N}} = x^{e;\mathbf{N}} \mid I^{s;\mathbf{N}} = \mathbf{i}^{s;\mathbf{N}}, I^{e;\mathbf{D}} = \mathbf{i}^{e;\mathbf{N}}) \\
 & \propto P(x^{s;\mathbf{N}} \mid \mathbf{i}^{s;\mathbf{N}}) P(x^{e;\mathbf{N}} \mid \mathbf{i}^{e;\mathbf{N}}) \frac{P(x^{s;\mathbf{N}}, x^{e;\mathbf{N}})}{P(x^{s;\mathbf{N}}) P(x^{e;\mathbf{N}})}. \tag{5.6}
 \end{aligned}$$

The HSBN<sup>non-dominant</sup> model in Figure 5.3 yields the following decomposition of the prior probability distributions for the observed handshape labels  $x^{s;\mathbf{N}}$ ,  $x^{e;\mathbf{N}}$ :

$$P(x^{s;\mathbf{N}}, x^{e;\mathbf{N}}) = \sum_{z \in \mathcal{Z}^{\mathbf{N}}} \pi_z^{\mathbf{N}} \mathbf{b}_z^{\mathbf{N}}(x^{s;\mathbf{N}}) \mathbf{b}_z^{\mathbf{N}}(x^{e;\mathbf{N}}). \tag{5.7}$$

Computing the marginals and substituting into Equation 5.6 yields the required posterior distributions for handshape labels on the non-dominant hand.

## 5.4 Summary

In this chapter we have described the HSBN graphical model for the task of start/end handshape inference in monomorphemic lexical signs. We proposed different adaptations of the HSBN model to accommodate the properties of articulation that are specific to one-handed and two-handed signs. The HSBN includes a hidden layer of random variables in order to model the properties of sign-independent phonological variation attested in handshape articulation. Given the HSBN model parameters, closed form expressions for posterior distributions over handshape labels are obtained and therefore the algorithm for handshape inference is computationally straightforward. Handshape inference using the HSBN produces a ranked list of candidate handshape labels and thereby facilitates the integration of handshape inference results with other computer vision based components towards developing a full-fledged system for sign recognition and retrieval.

## Chapter 6

# Learning the HSBN model

Given the HSBN model developed in the previous chapter, we now formulate a supervised learning framework for estimating the model parameters. We need to estimate the state-space  $\mathbf{Z} = (\mathcal{Z}^s, \mathcal{Z}^e)$  for representing hidden variables and the parameters  $\lambda_{\mathbf{Z}} = \{\boldsymbol{\pi}, \mathbf{a}, \mathbf{b}^s, \mathbf{b}^e\}$  for the multinomial distributions. In this chapter we develop an approach to estimate the multinomial parameters assuming that a state-space for the hidden variables is available. In the next chapter we develop an algorithm to explore the state-space in order to determine a suitable representation for the hidden variables. We rely on the variational Bayes formulation [Beal, 2003] in developing the learning algorithms for these two parameter estimation tasks.

The training set contains productions of monomorphemic lexical signs from a vocabulary  $\mathcal{V}_{\mathbf{x}}$ . The training set is assumed to include examples produced by two or more native sign language users for a significant fraction of items in the vocabulary. The availability of multiple productions of a sign allows the learning algorithm to accrue patterns of sign-independent variability in handshape articulation. The productions of signs in the training set are assumed to be annotated with their start/end handshape labels.

The handshape label annotations in one-handed signs are depicted as  $\mathbf{x}_{ij} = (x_{ij}^s, x_{ij}^e)$ , where,  $x_{ij}^s, x_{ij}^e \in \mathcal{X}$  are the start and end handshapes respectively. Here,  $i$  ranges over the items in the vocabulary, i.e.,  $1 \leq i \leq |\mathcal{V}_{\mathbf{x}}|$ , and  $j$  ranges over the different productions of the  $i^{\text{th}}$  vocabulary item.  $\mathcal{X}$  represents the set of all handshape labels. The handshapes annotated in two-handed signs are depicted by the tuple  $\mathbf{x}_{ij} = (x_{ij}^{s;\text{D}}, x_{ij}^{s;\text{N}}, x_{ij}^{e;\text{D}}, x_{ij}^{e;\text{N}})$ ; the superscripts D and N refer to handshapes articulated on the dominant and non-dominant hands respectively. The set  $\mathbf{x}_i = \{\mathbf{x}_{ij}\}$  refers to the handshape tuples for all examples

Notation	Description
$\mathcal{V}_{\mathbf{x}}$	The vocabulary of distinct monomorphemic lexical signs contained in a given training set
$\mathbf{x}, \mathbf{x}_i, \mathbf{x}_{ij},$ $(x_{ij}^s, x_{ij}^e), \mathcal{X}$	Start/end handshape label annotations for productions of signs contained in the training set, $\mathbf{x} = \{\mathbf{x}_i\}, 1 \leq i \leq  \mathcal{V}_{\mathbf{x}} , \quad \mathbf{x}_i = \{\mathbf{x}_{ij}\}, 1 \leq j \leq  \mathbf{x}_i ,$ $\mathbf{x}_{ij} = (x_{ij}^s, x_{ij}^e), \quad x_{ij}^s, x_{ij}^e \in \mathcal{X}$
$\mathbf{z}, \mathbf{z}_i, (z_i^s, z_i^e),$ $(\mathcal{Z}^s, \mathcal{Z}^e), \mathbf{Z}$	Hidden variables associated with signs in the vocabulary, $\mathbf{z} = \{\mathbf{z}_i\}, 1 \leq i \leq  \mathcal{V}_{\mathbf{x}} ,$ $\mathbf{z}_i = (z_i^s, z_i^e), \quad z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e, \quad \mathbf{Z} = (\mathcal{Z}^s, \mathcal{Z}^e)$

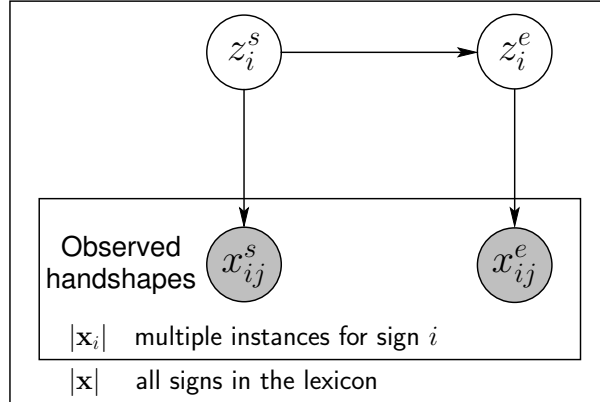
**Table 6.1:** Notations for the training set and hidden variables employed in learning the HSBN<sup>dominant</sup> model.

of the  $i^{\text{th}}$  vocabulary item and the set  $\mathbf{x} = \{\mathbf{x}_i\}$  refers to the handshape tuples for all vocabulary items contained in the training set. A summary of symbols for the training set are given in Table 6.1.

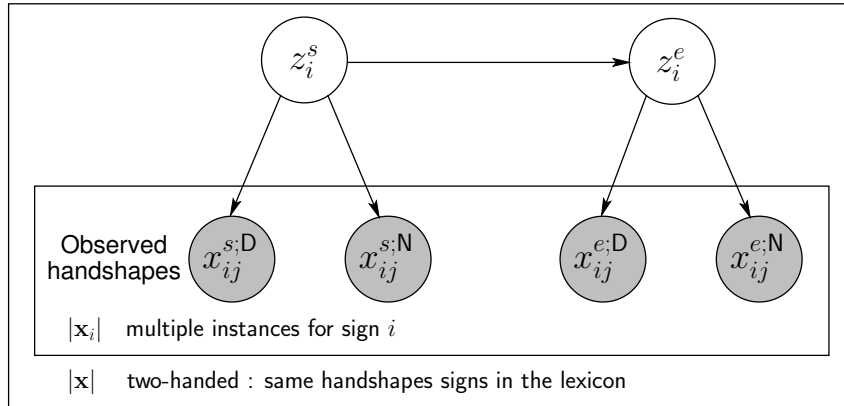
The HSBN model includes unobserved start/end hidden variables. The start/end handshape labels for the tuples contained in  $\mathbf{x}_i$  are assumed to arise as different realizations of a single pair of hidden variables,  $\mathbf{z}_i = (z_i^s, z_i^e)$ . The collection of hidden variable pairs for all signs in the vocabulary is represented as  $\mathbf{z} = \{\mathbf{z}_i\}, 1 \leq i \leq |\mathcal{V}_{\mathbf{x}}|$ . The hidden variables  $(z_i^s, z_i^e)$  take values from their respective state-spaces  $(\mathcal{Z}^s, \mathcal{Z}^e)$ . The one-to-many associations between the hidden variables and the handshape label annotations for the case of one-handed signs in the training set can be depicted using the plate representation as in Figure 6.1. An extension of this representation involving both one-handed and two-handed: same handshapes signs is depicted in Figure 6.2.

Given the state-space  $\mathbf{Z}$  for hidden variables, there is a choice among the different learning methods that can be adopted for parameter estimation. Here we consider the Maximum A-Posteriori (MAP) approach and its extension the variational Bayes (VB) approach. The VB formulation yields a lower bound,  $L_{\mathbf{Z}}^{\text{VB}}$ , that will prove instrumental in formulating an approach for state-space estimation.





**Figure 6-1:** A plate representation for one-handed signs contained in the training set is used to depict the one-to-many associations between the hidden variables and the annotated start/end handshape labels that are utilized in learning parameters involving hidden variables in the  $\text{HSBN}^{\text{dominant}}$  model.



**Figure 6-2:** A plate representation of the training set consisting of start/end handshape labels for two-handed : same handshapes signs is employed in learning parameters for the  $\text{HSBN}^{\text{congruent}}$  model.

We first develop the MAP and VB approaches for the case of one-handed signs. These learning formulations are subsequently extended to also include two-handed : same handshapes signs.

### 6.1 The MAPEM formulation for learning $\text{HSBN}^{\text{dominant}}$ model parameters

We present the MAP (Maximum A-Posteriori) formulation to learn the HSNB model parameters,  $\lambda$ . The inputs given are a training set,  $\mathbf{x}$ , and the prior distributions for the model

Notation	Description
$\boldsymbol{\pi}_{z^s}$ or $\boldsymbol{\pi}[z^s]$	The prior distribution $P(Z^s = z^s)$ , for the hidden state at the start of a sign
$\mathbf{a}_{z^s, z^e}$ or $\mathbf{a}[z^s, z^e]$	Transition probabilities $P(Z^e = z^e   Z^s = z^s)$ for start/end hidden states
$\mathbf{b}_{z^s}^s(x^s)$ or $\mathbf{b}^s[z^s, x^s]$ , $\mathbf{b}_{z^e}^e(x^e)$ or $\mathbf{b}^e[z^e, x^e]$	The probabilities for observed handshape labels to be obtained as different realizations of hidden states: $P(X^s = x^s   Z^s = z^s)$ , $P(X^e = x^e   Z^e = z^e)$
$\boldsymbol{\lambda}$	The parameters $\{\boldsymbol{\pi}, \mathbf{a}, \mathbf{b}^s, \mathbf{b}^e\}$ for the HSBN model
$\boldsymbol{\omega} = \{\boldsymbol{\nu}, \boldsymbol{\alpha}, \boldsymbol{\beta}^s, \boldsymbol{\beta}^e\}$	Dirichlet distribution parameters (hyper-parameters) associated with the HSBN parameters $\boldsymbol{\lambda} = \{\boldsymbol{\pi}, \mathbf{a}, \mathbf{b}^s, \mathbf{b}^e\}$ . These are defined as follows, $\boldsymbol{\nu}_{z^s}$ or $\boldsymbol{\nu}[z^s]$ is associated with $\boldsymbol{\pi}_{z^s}$ or $\boldsymbol{\pi}[z^s]$ $\boldsymbol{\alpha}_{z^s, z^e}$ or $\boldsymbol{\alpha}[z^s, z^e]$ is associated with $\mathbf{a}_{z^s, z^e}$ or $\mathbf{a}[z^s, z^e]$ $\boldsymbol{\beta}_{z^s}^s(x^s)$ or $\boldsymbol{\beta}^s[z^s, x^s]$ is associated with $\mathbf{b}_{z^s}^s(x^s)$ or $\mathbf{b}^s[z^s, x^s]$ $\boldsymbol{\beta}_{z^e}^e(x^e)$ or $\boldsymbol{\beta}^e[z^e, x^e]$ is associated with $\mathbf{b}_{z^e}^e(x^e)$ or $\mathbf{b}^e[z^e, x^e]$

**Table 6.2:** Parameters in the HSBN learning formulation.

parameters. The priors belong to the Dirichlet family whose (hyper)parameters are  $\boldsymbol{\omega}$ . The parameters are summarized in Table 6.2. In the MAP formulation we aim to maximize the posterior distribution over model parameters to yield an estimate for the model parameters:

$$\hat{\boldsymbol{\lambda}} = \arg \max_{\boldsymbol{\lambda}} [\ln P(\boldsymbol{\lambda} | \mathbf{x}, \boldsymbol{\omega}^{\text{prior}})]. \quad (6.1)$$

The exact posterior log-likelihood is intractable to optimize directly because the HSBN involves unobserved hidden variables  $\mathbf{z}$ . A lower bound to the posterior log-likelihood,  $L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda} | \mathbf{x}, Q_{\mathbf{z}})$ , is therefore constructed by introducing variational distributions  $Q_{\mathbf{z}}(\mathbf{z})$  for the hidden variables. Maximizing this lower bound using the Expectation-Maximization algorithm yields an estimate for the desired model parameters,  $\boldsymbol{\lambda}_{\mathbf{Z}}^{\text{MAP}}$ .

An expression for the MAPEM lower bound is formulated as follows,

$$\ln P(\boldsymbol{\lambda} | \mathbf{x}, \boldsymbol{\omega}^{\text{prior}}) = \ln P(\mathbf{x} | \boldsymbol{\lambda}, \boldsymbol{\omega}^{\text{prior}}) + \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}) - \ln P(\mathbf{x} | \boldsymbol{\omega}^{\text{prior}}) \quad (6.2)$$

$$\equiv \ln P(\mathbf{x} | \boldsymbol{\lambda}) + \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}) \quad (6.3)$$

$$= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \ln P(\mathbf{x}_i | \boldsymbol{\lambda}) + \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}) \quad (6.4)$$

$$= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \ln \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) + \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}). \quad (6.5)$$

Equation 6.2 is obtained by applying Bayes' rule and is simplified in Equation 6.3 because  $P(\mathbf{x} | \boldsymbol{\omega}^{\text{prior}})$  is a constant for the given training set. The decomposition in Equation 6.4 is obtained because the handshape tuples,  $\mathbf{x}_i$ , for different vocabulary items are conditionally independent given the model parameters. Equation 6.5 introduces the pair of hidden variables,  $z_i^s, z_i^e$ , for each vocabulary item and marginalizes over them. We include the subscript  $i$  when required for clarity to denote that a pair of hidden variables is associated with a specific vocabulary item. These hidden variables take values from the corresponding state-spaces ( $\mathcal{Z}^s, \mathcal{Z}^e$ ). The marginalization is therefore performed over all settings of all hidden variables.

Variational distributions,  $Q_{\mathbf{z}}(\mathbf{z}) = \{Q_{\mathbf{z},i}(z_i^s, z_i^e)\}$ ,  $1 \leq i \leq |\mathcal{V}_{\mathbf{x}}|$ , are now introduced to yield a lower bound by allowing the log operator to be shifted inside the summation.

$$\ln P(\boldsymbol{\lambda} | \mathbf{x}, \boldsymbol{\omega}^{\text{prior}}) \equiv \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \ln \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \frac{P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})}{Q_{\mathbf{z},i}(z_i^s, z_i^e)} + \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}), \quad (6.6)$$

where the variational distributions have the following constraints,

$$\sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) = 1, \quad Q_{\mathbf{z},i}(z_i^s, z_i^e) \geq 0. \quad (6.7)$$

We will also define the following marginals for the variational distributions,

$$Q_{\mathbf{z},i}(z_i^s) = \sum_{z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e), \quad Q_{\mathbf{z},i}(z_i^e) = \sum_{z_i^s \in \mathcal{Z}^s} Q_{\mathbf{z},i}(z_i^s, z_i^e). \quad (6.8)$$

Using Jensen's inequality we obtain the desired MAPEM lower bound as follows,

$$\ln P(\boldsymbol{\lambda} | \mathbf{x}, \boldsymbol{\omega}^{\text{prior}}) \geq \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \ln \frac{P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})}{Q_{\mathbf{z},i}(z_i^s, z_i^e)} + \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}) \quad (6.9)$$

$$\begin{aligned} &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) [\ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e)] \\ &\quad + \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}) \end{aligned} \quad (6.10)$$

$$= L_{\mathcal{Z}}^{\text{MAP}}(\boldsymbol{\lambda} | \mathbf{x}, Q_{\mathbf{z}}). \quad (6.11)$$

The complete data log-likelihood term,  $\ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})$ , in Equation 6.10 can be expanded given the plate representation for the training set depicted in Figure 6-1,

$$\ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) = \ln \boldsymbol{\pi}_{z_i^s} + \ln \mathbf{a}_{z_i^s, z_i^e} + \sum_{j=1}^{|\mathbf{x}_i|} [\ln \mathbf{b}_{z_i^s}^s(x_{ij}^s) + \ln \mathbf{b}_{z_i^e}^e(x_{ij}^e)]. \quad (6.12)$$

The priors for model parameters,  $P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}})$ , in Equation 6.10 can be expanded as follows given,  $\boldsymbol{\omega}^{\text{prior}}$ , the hyper-parameters for Dirichlet priors associated with each model parameter,

$$\begin{aligned} \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}) &= \ln \text{Dir}(\boldsymbol{\pi} | \boldsymbol{\nu}^{\text{prior}}) + \sum_{z^s \in \mathcal{Z}^s} \ln \text{Dir}(\mathbf{a}_{z^s} | \boldsymbol{\alpha}_{z^s}^{\text{prior}}) \\ &\quad + \sum_{z^s \in \mathcal{Z}^s} \ln \text{Dir}(\mathbf{b}_{z^s}^s | \boldsymbol{\beta}_{z^s}^{s \text{ prior}}) + \sum_{z^e \in \mathcal{Z}^e} \ln \text{Dir}(\mathbf{b}_{z^e}^e | \boldsymbol{\beta}_{z^e}^{e \text{ prior}}) \end{aligned} \quad (6.13)$$

$$\begin{aligned} &= \sum_{z^s \in \mathcal{Z}^s} (\boldsymbol{\nu}_{z^s}^{\text{prior}} - 1) \ln \boldsymbol{\pi}_{z^s} + \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} (\boldsymbol{\alpha}_{z^s, z^e}^{\text{prior}} - 1) \ln \mathbf{a}_{z^s, z^e} \\ &\quad + \sum_{z^s \in \mathcal{Z}^s, x \in \mathcal{X}} (\boldsymbol{\beta}_{z^s}^{s \text{ prior}}(x) - 1) \ln \mathbf{b}_{z^s}^s(x) \\ &\quad + \sum_{z^e \in \mathcal{Z}^e, x \in \mathcal{X}} (\boldsymbol{\beta}_{z^e}^{e \text{ prior}}(x) - 1) \ln \mathbf{b}_{z^e}^e(x). \end{aligned} \quad (6.14)$$

With the above two terms in place, the overall objective for the MAPEM formulation is given by:

$$\max_{\boldsymbol{\lambda}, Q_{\mathbf{z}}} \left[ L_{\mathcal{Z}}^{\text{MAP}}(\boldsymbol{\lambda} | \mathbf{x}, Q_{\mathbf{z}}) \right]. \quad (6.15)$$

This lower bound is maximized using a block coordinate ascent approach [Dempster et al., 1977]. The maximization is performed in alternation with respect to the variational distributions,  $Q_{\mathbf{z}}$ , and the model parameters,  $\boldsymbol{\lambda}$ , to yield the updated values for the variational distributions and the model parameters respectively. These two maximization steps constitute the E and M steps in the MAPEM algorithm. The two key equations (Equation 6.19 and Equations 6.24-6.27) for learning the HSBN differ from those of the MAP formulation for HMMs by including the one-to-many associations between the hidden variables  $z_i^s, z_i^e$  and observed variables  $\mathbf{x}_i = \left\{ \left( x_{ij}^s, x_{ij}^e \right) \right\}$ .

The equations for the E-step, the M-step and the update for the MAPEM lower bound are derived below.

For the MAPEM E-step, we maximize the lower bound with respect to  $Q_{\mathbf{z}}$  while holding  $\boldsymbol{\lambda}$  constant:

$$\begin{aligned} & \max_{Q_{\mathbf{z}}} \left[ L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda} \mid \mathbf{x}, Q_{\mathbf{z}}) \right] \\ \text{Subject to: } & \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) = 1 \end{aligned} \quad (6.16)$$

The desired updates to the variational distributions are derived using Lagrange multipliers  $\mu_{Q_{\mathbf{z},i}}$  for the sum-to-one constraints,

$$\nabla_{Q_{\mathbf{z},i}}(L_{\mathbf{Z}}^{\text{MAP}}) + \mu_{Q_{\mathbf{z},i}} = 0.$$

Substituting the expression for  $L_{\mathbf{Z}}^{\text{MAP}}$  from Equation 6.10, we obtain,

$$\begin{aligned} \nabla_{Q_{\mathbf{z},i}} \left[ \sum_{i=1}^{|\mathcal{X}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) [\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \boldsymbol{\lambda}) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e)] + \ln P(\boldsymbol{\lambda} \mid \boldsymbol{\omega}^{\text{prior}}) \right] + \mu_{Q_{\mathbf{z},i}} &= 0 \\ \therefore \nabla_{Q_{\mathbf{z},i}} [Q_{\mathbf{z},i}(z_i^s, z_i^e) (\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \boldsymbol{\lambda}) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e))] + \mu_{Q_{\mathbf{z},i}} &= 0 \\ \therefore \frac{\partial}{\partial Q_{\mathbf{z},i}(z_i^s, z_i^e)} [Q_{\mathbf{z},i}(z_i^s, z_i^e) (\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \boldsymbol{\lambda}) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e))] + \mu_{Q_{\mathbf{z},i}} &= 0 \\ \therefore \ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \boldsymbol{\lambda}) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) - 1 + \mu_{Q_{\mathbf{z},i}} &= 0. \end{aligned} \quad (6.17)$$

The variational distributions can therefore be expressed as,

$$\ln Q_{\mathbf{z},i}(z_i^s, z_i^e) = \ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) - C_{Q_{\mathbf{z},i}}, \quad (6.18)$$

where the  $C_{Q_{\mathbf{z},i}}$  are normalizing constants for the variational distributions  $Q_{\mathbf{z},i}$ .

Substituting from Equation 6.12 we obtain an expression for variational distributions associated with the hidden variables, thereby concluding the derivation of the MAPEM E-step,

$$\ln Q_{\mathbf{z},i}(z_i^s, z_i^e) = \ln \pi_{z_i^s} + \ln \mathbf{a}_{z_i^s, z_i^e} + \sum_{j=1}^{|\mathbf{x}_i|} \left[ \ln \mathbf{b}_{z_i^s}^s(x_{ij}^s) + \ln \mathbf{b}_{z_i^e}^e(x_{ij}^e) \right] - C_{Q_{\mathbf{z},i}}. \quad (6.19)$$

For the MAPEM M-step, we maximize the lower bound with respect to  $\boldsymbol{\lambda}$  while holding the variational distributions  $Q_{\mathbf{z}}(\mathbf{z})$  constant:

$$\max_{\boldsymbol{\lambda}} \left[ L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda} | \mathbf{x}, Q_{\mathbf{z}}) \right]$$

Subject to: The stochasticity constraints on  $\boldsymbol{\lambda}$  given by,

$$\begin{aligned} \sum_{z^s \in \mathcal{Z}^s} \pi_{z^s} &= 1; & \sum_{z^e \in \mathcal{Z}^e} \mathbf{a}_{z^s, z^e} &= 1, \forall z^s \in \mathcal{Z}^s, \\ \sum_{x \in \mathcal{X}} \mathbf{b}_{z^s}^s(x) &= 1, \forall z^s \in \mathcal{Z}^s; & \sum_{x \in \mathcal{X}} \mathbf{b}_{z^e}^e(x) &= 1, \forall z^e \in \mathcal{Z}^e. \end{aligned} \quad (6.20)$$

Introducing Lagrange multipliers  $\mu$  for the above constraints, we obtain,

$$\begin{aligned} \nabla_{\boldsymbol{\pi}}(L_{\mathbf{Z}}^{\text{MAP}}) + \mu_{\boldsymbol{\pi}} &= 0, & \nabla_{\mathbf{a}_{z^s}}(L_{\mathbf{Z}}^{\text{MAP}}) + \mu_{\mathbf{a}_{z^s}} &= 0 \\ \nabla_{\mathbf{b}_{z^s}^s}(L_{\mathbf{Z}}^{\text{MAP}}) + \mu_{\mathbf{b}_{z^s}^s} &= 0, & \nabla_{\mathbf{b}_{z^e}^e}(L_{\mathbf{Z}}^{\text{MAP}}) + \mu_{\mathbf{b}_{z^e}^e} &= 0. \end{aligned} \quad (6.21)$$

From Equation 6.10, we have,

$$L_{\mathbf{Z}}^{\text{MAP}} = \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \left[ \ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) \right] + \ln P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}}). \quad (6.22)$$

Substituting the expression for  $P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})$  from Equation 6.12 and the expression for

$P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}})$  from Equation 6.14 in the above equation we obtain,

$$\begin{aligned}
L_{\mathbf{Z}}^{\text{MAP}} &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \left[ \ln \pi_{z_i^s} + \ln \mathbf{a}_{z_i^s, z_i^e} + \sum_{j=1}^{|\mathbf{x}_i|} \left[ \ln \mathbf{b}_{z_i^s}^s(x_{ij}^s) + \ln \mathbf{b}_{z_i^e}^e(x_{ij}^e) \right] \right] \\
&\quad - \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) \\
&\quad + \sum_{z^s \in \mathcal{Z}^s} \left( \nu_{z^s}^{\text{prior}} - 1 \right) \ln \pi_{z^s} + \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} \left( \alpha_{z^s, z^e}^{\text{prior}} - 1 \right) \ln \mathbf{a}_{z^s, z^e} \\
&\quad + \sum_{z^s \in \mathcal{Z}^s, x \in \mathcal{X}} \left( \beta_{z^s}^{\text{prior}}(x) - 1 \right) \ln \mathbf{b}_{z^s}^s(x) \\
&\quad + \sum_{z^e \in \mathcal{Z}^e, x \in \mathcal{X}} \left( \beta_{z^e}^{\text{prior}}(x) - 1 \right) \ln \mathbf{b}_{z^e}^e(x). \tag{6.23}
\end{aligned}$$

Setting the derivatives of the above expression with respect to each of the model parameters to 0 yields the desired updates for the model parameters (Equations 6.24-6.27).

The updated values for  $\boldsymbol{\pi}^*$  are given by,

$$\begin{aligned}
\frac{\partial}{\partial \pi_{z^s}} \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \ln \pi_{z_i^s} + \sum_{\tilde{z}^s \in \mathcal{Z}^s} \left( \nu_{\tilde{z}^s}^{\text{prior}} - 1 \right) \ln \pi_{\tilde{z}^s} \right] + \mu_{\boldsymbol{\pi}} &= 0 \\
\therefore \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \delta(z^s, z_i^s) \frac{1}{\pi_{z_i^s}} & \\
&\quad + \sum_{\tilde{z}^s \in \mathcal{Z}^s} \left( \nu_{\tilde{z}^s}^{\text{prior}} - 1 \right) \delta(z^s, \tilde{z}^s) \frac{1}{\pi_{\tilde{z}^s}} + \mu_{\boldsymbol{\pi}} = 0 \\
\therefore \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \frac{1}{\pi_{z_i^s}} + \left( \nu_{z^s}^{\text{prior}} - 1 \right) \frac{1}{\pi_{z^s}} + \mu_{\boldsymbol{\pi}} &= 0 \\
\therefore \pi_{z^s}^* = \frac{1}{C_{\boldsymbol{\pi}}} \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s) + \left( \nu_{z^s}^{\text{prior}} - 1 \right) \right] \forall z^s \in \mathcal{Z}^s. &\tag{6.24}
\end{aligned}$$

The updated values for  $\mathbf{a}^*$  are given by,

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{a}_{z^s, z^e}} \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z}, i}(z_i^s, z_i^e) \ln \mathbf{a}_{z_i^s, z_i^e} \right. \\
& \quad \left. + \sum_{\tilde{z}^s \in \mathcal{Z}^s, \tilde{z}^e \in \mathcal{Z}^e} \left( \alpha_{\tilde{z}^s, \tilde{z}^e}^{\text{prior}} - 1 \right) \ln \mathbf{a}_{\tilde{z}^s, \tilde{z}^e} \right] + \mu_{\mathbf{a}_{z^s}} = 0 \\
& \therefore \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z}, i}(z_i^s, z_i^e) \delta(z^s, z_i^s) \delta(z^e, z_i^e) \frac{1}{\mathbf{a}_{z_i^s, z_i^e}} \\
& \quad + \sum_{\tilde{z}^s \in \mathcal{Z}^s, \tilde{z}^e \in \mathcal{Z}^e} \left( \alpha_{\tilde{z}^s, \tilde{z}^e}^{\text{prior}} - 1 \right) \delta(z^s, \tilde{z}^s) \delta(z^e, \tilde{z}^e) \frac{1}{\mathbf{a}_{\tilde{z}^s, \tilde{z}^e}} + \mu_{\mathbf{a}_{z^s}} = 0 \\
& \therefore \mathbf{a}_{z^s, z^e}^* = \frac{1}{C_{\mathbf{a}_{z^s}}} \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z}, i}(z^s, z^e) + \left( \alpha_{z^s, z^e}^{\text{prior}} - 1 \right) \right] \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e. \quad (6.25)
\end{aligned}$$

The updated values for  $\mathbf{b}^{s*}$  and  $\mathbf{b}^{e*}$  are given by,

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{b}_{z^s}^s(x)} \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z}, i}(z_i^s, z_i^e) \sum_{j=1}^{|\mathbf{x}_i|} \ln \mathbf{b}_{z_i^s}^s(x_{ij}^s) \right. \\
& \quad \left. + \sum_{\tilde{z}^s \in \mathcal{Z}^s, \tilde{x} \in \mathcal{X}} \left( \beta_{\tilde{z}^s}^{\text{prior}}(\tilde{x}) - 1 \right) \ln \mathbf{b}_{\tilde{z}^s}^s(\tilde{x}) \right] + \mu_{\mathbf{b}_{z^s}^s} = 0 \\
& \therefore \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z}, i}(z_i^s, z_i^e) \sum_{j=1}^{|\mathbf{x}_i|} \delta(z^s, z_i^s) \delta(x, x_{ij}^s) \frac{1}{\mathbf{b}_{z_i^s}^s(x_{ij}^s)} \\
& \quad + \sum_{\tilde{z}^s \in \mathcal{Z}^s, \tilde{x} \in \mathcal{X}} \left( \beta_{\tilde{z}^s}^{\text{prior}}(\tilde{x}) - 1 \right) \delta(z^s, \tilde{z}^s) \delta(x, \tilde{x}) \frac{1}{\mathbf{b}_{\tilde{z}^s}^s(\tilde{x})} + \mu_{\mathbf{b}_{z^s}^s} = 0 \\
& \therefore \mathbf{b}_{z^s}^{s*}(x) = \frac{1}{C_{\mathbf{b}_{z^s}^s}} \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z}, i}(z^s) \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^s) + \left( \beta_{z^s}^{\text{prior}}(x) - 1 \right) \right] \forall z^s \in \mathcal{Z}^s, \quad (6.26)
\end{aligned}$$

$$\begin{aligned}
& \mathbf{b}_{z^e}^{e*}(x) = \frac{1}{C_{\mathbf{b}_{z^e}^e}} \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z}, i}(z^e) \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^e) + \left( \beta_{z^e}^{\text{prior}}(x) - 1 \right) \right] \forall z^e \in \mathcal{Z}^e. \quad (6.27)
\end{aligned}$$

The scalars  $C_{\pi}$ ,  $C_{\mathbf{a}_{z^s}}$ ,  $C_{\mathbf{b}_{z^s}^s}$  and  $C_{\mathbf{b}_{z^e}^e}$  are normalizing constants to satisfy stochasticity constraints for the multinomial parameters.



The updated value for the MAPEM lower bound can now be derived as follows. From Equation 6.10 we have,

$$L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda} \mid \mathbf{x}, Q_{\mathbf{z}}) = \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) [\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \boldsymbol{\lambda}) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e)] + \ln P(\boldsymbol{\lambda} \mid \boldsymbol{\omega}^{\text{prior}}) \quad (6.28)$$

Substituting the expression for  $\ln Q_{\mathbf{z},i}(z_i^s, z_i^e)$  from Equation 6.18, we get,

$$\begin{aligned} L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda} \mid \mathbf{x}, Q_{\mathbf{z}}) &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) [C_{Q_{\mathbf{z},i}}] + \ln P(\boldsymbol{\lambda} \mid \boldsymbol{\omega}^{\text{prior}}) \\ &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} C_{Q_{\mathbf{z},i}} + \ln P(\boldsymbol{\lambda} \mid \boldsymbol{\omega}^{\text{prior}}). \end{aligned} \quad (6.29)$$

In the second step above we used the property that the variational distributions are normalized:  $\sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) = 1$ .

Substituting the terms for  $\ln P(\boldsymbol{\lambda} \mid \boldsymbol{\omega}^{\text{prior}})$  from Equation 6.14, we get,

$$\begin{aligned} L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda}^* \mid \mathbf{x}, Q_{\mathbf{z}}) &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} C_{Q_{\mathbf{z},i}} + \sum_{z^s \in \mathcal{Z}^s} (\nu_{z^s}^{\text{prior}} - 1) \ln \pi_{z^s}^* \\ &\quad + \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} (\alpha_{z^s, z^e}^{\text{prior}} - 1) \ln \mathbf{a}_{z^s, z^e}^* \\ &\quad + \sum_{z^s \in \mathcal{Z}^s, x \in \mathcal{X}} (\beta_{z^s}^{\text{prior}}(x) - 1) \ln \mathbf{b}_{z^s}^*(x) \\ &\quad + \sum_{z^e \in \mathcal{Z}^e, x \in \mathcal{X}} (\beta_{z^e}^{\text{prior}}(x) - 1) \ln \mathbf{b}_{z^e}^*(x). \end{aligned} \quad (6.30)$$

The complete MAPEM algorithm is summarized in Algorithm 6.1. The inputs given are the training set  $\mathbf{x}$  containing the start/end handshape labels for signs in a vocabulary  $\mathcal{V}_{\mathbf{x}}$ , the initial value for model parameters  $\boldsymbol{\lambda}^{\circ}$ , and the (hyper)parameters  $\boldsymbol{\omega}^{\text{prior}}$  for the Dirichlet distributions that serve as the priors for the model parameters. The outputs from the algorithm are the MAP estimates for model parameters,  $\boldsymbol{\lambda}_{\mathbf{Z}}^{\text{MAP}}$ .

The model parameters are initialized in Algorithm 6.1, step 1. The E and M steps are used in alternation until the estimated parameters and the MAPEM lower bound converge. For the E-step, an estimate for the variational distributions,  $\tilde{Q}_{\mathbf{z},i}$ , for each item in the

vocabulary are computed as in Algorithm 6.1, step 4. The variational distributions are then normalized (Algorithm 6.1, step 5) and the normalizing constant is saved (as  $C_{Q_{\mathbf{z}}}[i]$ ). These constants are used in Algorithm 6.1, step 6 to compute the MAPEM lower bound  $L_{\mathbf{Z}}^{\text{MAP}}$ . The normalized variational distributions,  $Q_{\mathbf{z},i}$ , are used in the M-step (Algorithm 6.1, steps 7-10) to obtain an update for the model parameters.

## 6.2 Variational Bayes formulation for learning **HSBN<sup>dominant</sup>** model parameters

The variational Bayes (VB) approach employs a lower bound  $L_{\mathbf{Z}}^{\text{VB}}$  to the posterior likelihood  $P(\mathbf{x})$  for the given training set,  $\mathbf{x}$ . This is needed since the complete data-likelihood is intractable to compute directly: the hidden parameters introduce dependencies between latent variables associated with different training samples. Through the process of maximizing this lower bound, the VB approach yields an approximation to the desired posterior distribution over model parameters  $P(\boldsymbol{\lambda} | \mathbf{x})$ . Choosing Dirichlet priors with parameters  $\boldsymbol{\omega}^{\text{prior}} = \{\nu^{\text{prior}}, \boldsymbol{\alpha}^{\text{prior}}, \boldsymbol{\beta}^{s \text{ prior}}, \boldsymbol{\beta}^{e \text{ prior}}\}$  for the multinomial distributions in the HSBN model yields posterior distributions from the same family, denoted with parameters  $\boldsymbol{\omega}^* = \{\nu^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^{s*}, \boldsymbol{\beta}^{e*}\}$ .

An expression for the VB lower bound is obtained by introducing two sets of variational distributions,  $Q_{\mathbf{z},i}$ , for hidden variables  $\mathbf{z}_i$  and,  $Q_{\boldsymbol{\lambda}}$ , for model parameters  $\boldsymbol{\lambda}$ ,

$$\ln P(\mathbf{x}) = \ln \int d\boldsymbol{\lambda} P(\mathbf{x} | \boldsymbol{\lambda}) P(\boldsymbol{\lambda}) \quad (6.31)$$

$$= \ln \int d\boldsymbol{\lambda} Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) P(\mathbf{x} | \boldsymbol{\lambda}) \frac{P(\boldsymbol{\lambda})}{Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})}. \quad (6.32)$$

Variational distributions  $Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})$  associated with the model parameters  $\boldsymbol{\lambda}$  are introduced to

**Algorithm 6.1:** MAPEM<sup>dominant</sup> algorithm for learning HSNB<sup>dominant</sup> parameters

**Inputs** :  $\mathbf{x}$  Handshake label pairs for signs contained in a training set,  
:  $\lambda^\circ$  Initial values for the model parameters,  
:  $\omega^{\text{prior}}$  Prior distributions for HSNB parameters.

**Outputs**:  $\lambda_Z^{\text{MAP}}$  Estimated HSNB parameters.

- 1  $\lambda^* \leftarrow \lambda^\circ$ ;
- 2 **repeat**
  - /\* E-Step (derived in Equation 6.19) \*/
  - 3 **for**  $i \leftarrow 1$  to  $|\mathcal{V}_x|$  **do**
  - 4  $\tilde{Q}_{\mathbf{z},i}[z^s, z^e] \leftarrow$   

$$\exp \left[ \ln \pi^*[z^s] + \ln \mathbf{a}^*[z^s, z^e] + \sum_{j=1}^{|\mathbf{x}_i|} \ln \mathbf{b}^{s*}[z^s, x_{ij}^s] + \sum_{j=1}^{|\mathbf{x}_i|} \ln \mathbf{b}^{e*}[z^e, x_{ij}^e] \right] \quad \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$$
  - 5  $C_{Q_z}[i] \leftarrow \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} \tilde{Q}_{\mathbf{z},i}[z^s, z^e]; \quad Q_{\mathbf{z},i}[z^s, z^e] \leftarrow \frac{\tilde{Q}_{\mathbf{z},i}[z^s, z^e]}{C_{Q_z}[i]} \quad \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$
  - end**
  - /\* Update the MAPEM lower bound (derived in Equation 6.30) \*/
  - 6  $L_Z^{\text{MAP}} \leftarrow \sum_{i=1}^{|\mathcal{V}_x|} \ln C_{Q_z}[i] + \sum_{z^s \in \mathcal{Z}^s} (\nu^{\text{prior}}[z^s] - 1) \ln \pi^*[z^s] + \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} (\alpha^{\text{prior}}[z^s, z^e] - 1) \ln \mathbf{a}^*[z^s, z^e]$   
 $+ \sum_{z^s \in \mathcal{Z}^s, x \in \mathcal{X}} (\beta^s \text{prior}[z^s, x] - 1) \ln \mathbf{b}^{s*}[z^s, x] + \sum_{z^e \in \mathcal{Z}^e, x \in \mathcal{X}} (\beta^e \text{prior}[z^e, x] - 1) \ln \mathbf{b}^{e*}[z^e, x];$
  - /\* M-step (derived in Equations 6.24 - 6.27) \*/
  - 7  $\pi^*[z^s] \leftarrow \frac{1}{C_\pi} \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}[z^s, z^e] + (\nu^{\text{prior}}[z^s] - 1) \right] \quad \forall z^s \in \mathcal{Z}^s;$
  - 8  $\mathbf{a}^*[z^s, z^e] \leftarrow \frac{1}{C_{\mathbf{a}^s}} \left[ \sum_{i=1}^{|\mathcal{V}_x|} Q_{\mathbf{z},i}[z^s, z^e] + (\alpha^{\text{prior}}[z^s, z^e] - 1) \right] \quad \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$
  - 9  $\mathbf{b}^{s*}[z^s, x] \leftarrow \frac{1}{C_{\mathbf{b}_z^s}} \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}[z^s, z^e] \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^s) + (\beta^s \text{prior}[z^s, x] - 1) \right] \quad \forall z^s \in \mathcal{Z}^s, x \in \mathcal{X};$
  - 10  $\mathbf{b}^{e*}[z^e, x] \leftarrow \frac{1}{C_{\mathbf{b}_z^e}} \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^s \in \mathcal{Z}^s} Q_{\mathbf{z},i}[z^s, z^e] \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^e) + (\beta^e \text{prior}[z^e, x] - 1) \right] \quad \forall z^e \in \mathcal{Z}^e, x \in \mathcal{X};$
- until** the lower bound,  $L_Z^{\text{MAP}}$  **and** the parameters,  $\lambda^*$  converge;
- 11  $\lambda_Z^{\text{MAP}} \leftarrow \lambda^*$ ;

shift the log inside the integral using Jensen's inequality,

$$\begin{aligned}
\ln P(\mathbf{x}) &\geq \int d\boldsymbol{\lambda} Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) \ln P(\mathbf{x} | \boldsymbol{\lambda}) \frac{P(\boldsymbol{\lambda})}{Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})} & (6.33) \\
&= \int d\boldsymbol{\lambda} Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \ln P(\mathbf{x}_i | \boldsymbol{\lambda}) + \ln \frac{P(\boldsymbol{\lambda})}{Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})} \right] \\
&= \int d\boldsymbol{\lambda} Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \ln \sum_{z_i^s, z_i^e \in \mathcal{Z}^e} P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) + \ln \frac{P(\boldsymbol{\lambda})}{Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})} \right] \\
&= \int d\boldsymbol{\lambda} Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \ln \sum_{z_i^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \frac{P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})}{Q_{\mathbf{z},i}(z_i^s, z_i^e)} + \ln \frac{P(\boldsymbol{\lambda})}{Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})} \right]. & (6.34)
\end{aligned}$$

Variational distributions,  $Q_{\mathbf{z},i}(z_i^s, z_i^e)$ , for the hidden variables  $\mathbf{z}_i = (z_i^s, z_i^e)$  are introduced with the same properties as in Equation 6.7 to yield the following lower bound once again applying Jensen's inequality to shift the log operator inside the summation,

$$\ln P(\mathbf{x}) \geq \int d\boldsymbol{\lambda} Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \ln \frac{P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})}{Q_{\mathbf{z},i}(z_i^s, z_i^e)} + \ln \frac{P(\boldsymbol{\lambda})}{Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})} \right] & (6.35)$$

$$\begin{aligned}
&= \int d\boldsymbol{\lambda} Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \left[ \ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) \right] \right. \\
&\quad \left. + \ln P(\boldsymbol{\lambda}) - \ln Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) \right] & (6.36)
\end{aligned}$$

$$= L_{\mathcal{Z}}^{\text{VB}}(\mathbf{x} | Q_{\boldsymbol{\lambda}}, Q_{\mathbf{z}}). & (6.37)$$

The complete data log-likelihood term,  $\ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})$ , and the prior distribution for model parameters,  $P(\boldsymbol{\lambda} | \boldsymbol{\omega}^{\text{prior}})$ , are expanded as in the MAPEM formulation Equations 6.12 and 6.14.

The objective function for the VBEM formulation is therefore given by:

$$\max_{Q_{\boldsymbol{\lambda}}, Q_{\mathbf{z}}} \left[ L_{\mathcal{Z}}^{\text{VB}}(\mathbf{x} | Q_{\boldsymbol{\lambda}}, Q_{\mathbf{z}}) \right]. & (6.38)$$

This lower bound is maximized in alternation with respect to the two sets of variational distributions,  $Q_{\mathbf{z}}$  and  $Q_{\lambda}$ . These two updates constitute the E and M steps in the VBEM algorithm.

The equations for the M-step are derived first followed by the equations for the E-step. The equations to update the VB lower bound are then presented to conclude the VBEM formulation.

In the VBEM M-step we maximize the lower bound with respect to  $Q_{\lambda}$  while holding  $Q_{\mathbf{z}}(\mathbf{z})$  constant,

$$\begin{aligned} & \max_{Q_{\lambda}} \left[ L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x} \mid Q_{\lambda}, Q_{\mathbf{z}}) \right] \\ \text{Subject to: } & \int d\lambda Q_{\lambda}(\lambda) = 1. \end{aligned} \quad (6.39)$$

The expression for  $Q_{\lambda}(\lambda)$  obtained by optimizing the above objective approximates the desired posterior distributions,  $P(\lambda \mid \mathbf{x})$ . We proceed with this optimization using Lagrange multipliers,  $\mu_{Q_{\lambda}}$ ,

$$\nabla_{Q_{\lambda}}(L_{\mathbf{Z}}^{\text{VB}}) + \mu_{Q_{\lambda}} = 0. \quad (6.40)$$

Substituting the expression for  $L_{\mathbf{Z}}^{\text{VB}}$  from Equation 6.36, we get,

$$\begin{aligned} & \nabla_{Q_{\lambda}} \left( \int d\lambda Q_{\lambda}(\lambda) \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) [\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \lambda) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e)] \right. \right. \\ & \qquad \qquad \qquad \left. \left. + \ln P(\lambda) - \ln Q_{\lambda}(\lambda) \right] \right) + \mu_{Q_{\lambda}} = 0 \\ & \therefore \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) [\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \lambda) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e)] \right. \\ & \qquad \qquad \qquad \left. + \ln P(\lambda) - \ln Q_{\lambda}(\lambda) \right] - 1 + \mu_{Q_{\lambda}} = 0 \\ & \therefore \ln Q_{\lambda}(\lambda) = \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) [\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \lambda) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e)] \\ & \qquad \qquad \qquad + \ln P(\lambda) - 1 + \mu_{Q_{\lambda}}. \end{aligned} \quad (6.41)$$

Substituting the expression for  $\ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})$  from Equation 6.12, we obtain,

$$\begin{aligned} \ln Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \left[ \ln \pi_{z_i^s} + \ln \mathbf{a}_{z_i^s, z_i^e} + \sum_{j=1}^{|\mathbf{x}_i|} \left[ \ln \mathbf{b}_{z_i^s}^s(x_{ij}^s) + \ln \mathbf{b}_{z_i^e}^e(x_{ij}^e) \right] \right] \\ &\quad - \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) + \ln P(\boldsymbol{\lambda}) - 1 + \mu_{Q_{\boldsymbol{\lambda}}}. \end{aligned} \quad (6.42)$$

Substituting the expression for  $\ln P(\boldsymbol{\lambda})$  from Equation 6.14 and simplifying further,

$$\begin{aligned} \ln Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}) &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s} Q_{\mathbf{z},i}(z_i^s) \ln \pi_{z_i^s} + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \ln \mathbf{a}_{z_i^s, z_i^e} \\ &\quad + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s} Q_{\mathbf{z},i}(z_i^s) \sum_{j=1}^{|\mathbf{x}_i|} \ln \mathbf{b}_{z_i^s}^s(x_{ij}^s) \\ &\quad + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^e) \sum_{j=1}^{|\mathbf{x}_i|} \ln \mathbf{b}_{z_i^e}^e(x_{ij}^e) \\ &\quad + \sum_{z^s \in \mathcal{Z}^s} \left( \nu_{z^s}^{\text{prior}} - 1 \right) \ln \pi_{z^s} + \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} \left( \alpha_{z^s, z^e}^{\text{prior}} - 1 \right) \ln \mathbf{a}_{z^s, z^e} \\ &\quad + \sum_{z^s \in \mathcal{Z}^s, x \in \mathcal{X}} \left( \beta_{z^s}^{\text{prior}}(x) - 1 \right) \ln \mathbf{b}_{z^s}^s(x) \\ &\quad + \sum_{z^e \in \mathcal{Z}^e, x \in \mathcal{X}} \left( \beta_{z^e}^{\text{prior}}(x) - 1 \right) \ln \mathbf{b}_{z^e}^e(x) \\ &\quad + C_{Q_{\boldsymbol{\lambda}}}. \end{aligned} \quad (6.43)$$

Therefore,

$$\begin{aligned}
\ln Q_{\lambda}(\boldsymbol{\lambda}) &= \sum_{z^s \in \mathcal{Z}^s} \left( \boldsymbol{\nu}_{z^s}^{\text{prior}} + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s) - 1 \right) \ln \boldsymbol{\pi}_{z^s} \\
&+ \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} \left( \boldsymbol{\alpha}_{z^s, z^e}^{\text{prior}} + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s, z^e) - 1 \right) \ln \mathbf{a}_{z^s, z^e} \\
&+ \sum_{z^s \in \mathcal{Z}^s, x \in \mathcal{X}} \left( \boldsymbol{\beta}_{z^s}^{\text{prior}}(x) + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s) \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^s) - 1 \right) \ln \mathbf{b}_{z^s}^s(x) \\
&+ \sum_{z^e \in \mathcal{Z}^e, x \in \mathcal{X}} \left( \boldsymbol{\beta}_{z^e}^{\text{prior}}(x) + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^e) \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^e) - 1 \right) \ln \mathbf{b}_{z^e}^e(x) \\
&+ C_{Q_{\lambda}}, \tag{6.44}
\end{aligned}$$

where,  $C_{Q_{\lambda}}$  is a normalizing constant for the  $Q_{\lambda}$  variational distribution.

Further simplification yields the desired expression for the variational distributions  $Q_{\lambda}(\boldsymbol{\lambda})$  associated with the model parameters. The expression obtained below for  $Q_{\lambda}(\boldsymbol{\lambda})$  involves Dirichlet distributions with parameters  $\boldsymbol{\omega}^*$ . The update equations for these hyperparameters given in Equation 6.46 concludes the derivation for the M-step of the VBEM algorithm.

$$\begin{aligned}
\ln Q_{\lambda}(\boldsymbol{\lambda} | \boldsymbol{\omega}^*) &= \ln \text{Dir}(\boldsymbol{\pi} | \boldsymbol{\nu}^*) + \sum_{z^s \in \mathcal{Z}^s} \ln \text{Dir}(\mathbf{a}_{z^s} | \boldsymbol{\alpha}_{z^s}^*) \\
&+ \sum_{z^s \in \mathcal{Z}^s} \ln \text{Dir}(\mathbf{b}_{z^s}^s | \boldsymbol{\beta}_{z^s}^{s*}) + \sum_{z^e \in \mathcal{Z}^e} \ln \text{Dir}(\mathbf{b}_{z^e}^e | \boldsymbol{\beta}_{z^e}^{e*}). \tag{6.45}
\end{aligned}$$

A normalizing constant is not needed in Equation 6.45 because the RHS integrates to 1.

The updated hyper-parameters  $\omega^*$  in the above expression are as follows,

$$\begin{aligned}
\boldsymbol{\nu}_{z^s}^* &= \boldsymbol{\nu}_{z^s}^{\text{prior}} + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s), \\
\boldsymbol{\alpha}_{z^s, z^e}^* &= \boldsymbol{\alpha}_{z^s, z^e}^{\text{prior}} + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z_i^s, z_i^e), \\
\boldsymbol{\beta}_{z^s}^{s*}(x) &= \boldsymbol{\beta}_{z^s}^{s \text{ prior}}(x) + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s) \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^s), \\
\boldsymbol{\beta}_{z^e}^{e*}(x) &= \boldsymbol{\beta}_{z^e}^{e \text{ prior}}(x) + \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^e) \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^e),
\end{aligned} \tag{6.46}$$

where,  $Q_{\mathbf{z},i}(z^s)$  and  $Q_{\mathbf{z},i}(z^e)$  are the marginals described in Equation 6.8.

The variational distributions,  $Q_{\lambda}(\lambda | \omega^*)$  in Equation 6.45, approximate the desired posterior distribution over model parameters,  $P(\lambda | \mathbf{x}, \omega^{\text{prior}})$ . The mean for the estimated posterior distribution of model parameters is commonly employed as a point estimate for prediction given test inputs,

$$\boldsymbol{\lambda}_{\mathbf{Z}}^{\text{VB}} = \mathbb{E}_{Q_{\lambda}(\lambda | \omega^*)} [\boldsymbol{\lambda}]. \tag{6.47}$$

The expected values for the model parameters are obtained as follows,

$$\begin{aligned}
\boldsymbol{\pi}_{z^s}^* &= \frac{1}{C_{\boldsymbol{\pi}}} \boldsymbol{\nu}_{z^s}^* \quad \forall z^s \in \mathcal{Z}^s, \\
\mathbf{a}_{z^s, z^e}^* &= \frac{1}{C_{\mathbf{a}_{z^s}}} \boldsymbol{\alpha}_{z^s, z^e}^* \quad \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e, \\
\mathbf{b}_{z^s}^{s*}(x) &= \frac{1}{C_{\mathbf{b}_{z^s}^s}} \boldsymbol{\beta}_{z^s}^{s*}(x) \quad \forall z^s \in \mathcal{Z}^s, \\
\mathbf{b}_{z^e}^{e*}(x) &= \frac{1}{C_{\mathbf{b}_{z^e}^e}} \boldsymbol{\beta}_{z^e}^{e*}(x) \quad \forall z^e \in \mathcal{Z}^e.
\end{aligned} \tag{6.48}$$

The scalars  $C_{\boldsymbol{\pi}}$ ,  $C_{\mathbf{a}_{z^s}}$ ,  $C_{\mathbf{b}_{z^s}^s}$  and  $C_{\mathbf{b}_{z^e}^e}$  are normalizing constants to satisfy stochasticity constraints for the multinomial parameters.

In the VBEM E-step we maximize the lower bound with respect to  $Q_{\mathbf{z}}(\mathbf{z})$  while holding



$Q_\lambda$  constant,

$$\begin{aligned} & \max_{Q_{\mathbf{z}}} \left[ L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x} \mid Q_\lambda, Q_{\mathbf{z}}) \right] \\ \text{Subject to: } & \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) = 1. \end{aligned} \quad (6.49)$$

Using Lagrange multipliers  $\mu_{Q_{\mathbf{z},i}}$  we obtain the desired updates to the variational distributions:

$$\nabla_{Q_{\mathbf{z},i}}(L_{\mathbf{Z}}^{\text{VB}}) + \mu_{Q_{\mathbf{z},i}} = 0. \quad (6.50)$$

Substituting the expression for  $L_{\mathbf{Z}}^{\text{VB}}$  from Equation 6.36, we get,

$$\begin{aligned} \nabla_{Q_{\mathbf{z},i}} \left( \int d\lambda Q_\lambda(\lambda) \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) [\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \lambda) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e)] \right. \right. \\ \left. \left. + \ln P(\lambda) - \ln Q_\lambda(\lambda) \right] \right) + \mu_{Q_{\mathbf{z},i}} = 0 \\ \therefore \int d\lambda Q_\lambda(\lambda) [\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \lambda) - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) - 1] + \mu_{Q_{\mathbf{z},i}} = 0. \end{aligned} \quad (6.51)$$

The desired expressions for  $\ln Q_{\mathbf{z},i}(z_i^s, z_i^e)$  can now be derived as follows. Since  $\int d\lambda Q_\lambda(\lambda) = 1$ , we have,

$$\ln Q_{\mathbf{z},i}(z_i^s, z_i^e) = \int d\lambda Q_\lambda(\lambda) \ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \lambda) - C_{Q_{\mathbf{z},i}}. \quad (6.52)$$

The scalars  $C_{Q_{\mathbf{z},i}}$  are normalizing constants for the variational distributions,  $Q_{\mathbf{z},i}$ .

Using the expression for  $\ln P(\mathbf{x}_i, z_i^s, z_i^e \mid \lambda)$  from Equation 6.12 we get,

$$\begin{aligned} \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) = \int d\lambda Q_\lambda(\lambda) \left[ \ln \pi_{z_i^s} + \ln \mathbf{a}_{z_i^s, z_i^e} + \sum_{j=1}^{|\mathbf{x}_i|} \left[ \ln \mathbf{b}_{z_i^s}^s(x_{ij}^s) + \ln \mathbf{b}_{z_i^e}^e(x_{ij}^e) \right] \right] \\ - C_{Q_{\mathbf{z},i}}. \end{aligned} \quad (6.53)$$

Substituting the expression for  $Q_\lambda(\boldsymbol{\lambda})$  from Equation 6.45,

$$\begin{aligned} \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) &= \int d\boldsymbol{\pi} \operatorname{Dir}(\boldsymbol{\pi} | \boldsymbol{\nu}^*) \ln \pi_{z_i^s} + \int d\mathbf{a}_{z_i^s} \operatorname{Dir}(\mathbf{a}_{z_i^s} | \boldsymbol{\alpha}_{z_i^s}^*) \ln \mathbf{a}_{z_i^s, z_i^e} \\ &+ \sum_{j=1}^{|\mathbf{x}_i|} \int d\mathbf{b}_{z_i^s}^s \operatorname{Dir}(\mathbf{b}_{z_i^s}^s | \boldsymbol{\beta}_{z_i^s}^{s*}) \ln \mathbf{b}_{z_i^s}^s(x_{ij}^s) \\ &+ \sum_{j=1}^{|\mathbf{x}_i|} \int d\mathbf{b}_{z_i^e}^e \operatorname{Dir}(\mathbf{b}_{z_i^e}^e | \boldsymbol{\beta}_{z_i^e}^{e*}) \ln \mathbf{b}_{z_i^e}^e(x_{ij}^e) - C_{Q_{\mathbf{z},i}}. \end{aligned} \quad (6.54)$$

Using the identity

$$\int d\boldsymbol{\pi} \operatorname{Dir}(\boldsymbol{\pi} | \boldsymbol{\nu}) \ln \pi_i = \psi(\boldsymbol{\nu}_i) - \psi\left(\sum_k \boldsymbol{\nu}_k\right), \quad (6.55)$$

where  $\psi$  is the *digamma* function, we obtain,

$$\begin{aligned} \ln Q_{\mathbf{z},i}(z^s, z^e) &= -C_{Q_{\mathbf{z},i}} + \psi(\boldsymbol{\nu}_{z^s}^*) - \psi\left(\sum_{\tilde{z}^s \in \mathcal{Z}^s} \boldsymbol{\nu}_{\tilde{z}^s}^*\right) + \psi(\boldsymbol{\alpha}_{z^s, z^e}^*) - \psi\left(\sum_{\tilde{z}^e \in \mathcal{Z}^e} \boldsymbol{\alpha}_{\tilde{z}^e}^*\right) \\ &+ \sum_{j=1}^{|\mathbf{x}_i|} \left[ \psi(\boldsymbol{\beta}_{z^s}^{s*}(x_{ij}^s)) - \psi\left(\sum_{x \in \mathcal{X}} \boldsymbol{\beta}_{z^s}^{s*}(x)\right) + \psi(\boldsymbol{\beta}_{z^e}^{e*}(x_{ij}^e)) - \psi\left(\sum_{x \in \mathcal{X}} \boldsymbol{\beta}_{z^e}^{e*}(x)\right) \right]. \end{aligned} \quad (6.56)$$

The final step in the VBEM formulation is to obtain an expression for the updated value of the VB lower bound,  $L_{\mathbf{Z}}^{\text{VB}}$ . We use the expression for the VB lower bound in Equation 6.35,

$$\begin{aligned} L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x} | Q_\lambda, Q_{\mathbf{z}}) &= \int d\boldsymbol{\lambda} Q_\lambda(\boldsymbol{\lambda}) \left[ \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \ln \frac{P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda})}{Q_{\mathbf{z},i}(z_i^s, z_i^e)} + \ln \frac{P(\boldsymbol{\lambda})}{Q_\lambda(\boldsymbol{\lambda})} \right] \\ &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} \sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) \left[ \int d\boldsymbol{\lambda} Q_\lambda(\boldsymbol{\lambda}) \ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) \right. \\ &\quad \left. - \ln Q_{\mathbf{z},i}(z_i^s, z_i^e) \right] + \int d\boldsymbol{\lambda} Q_\lambda(\boldsymbol{\lambda}) \ln \frac{P(\boldsymbol{\lambda})}{Q_\lambda(\boldsymbol{\lambda})}. \end{aligned} \quad (6.57)$$

Substituting the expression for  $\ln Q_{\mathbf{z},i}(z_i^s, z_i^e)$  from the E-step (Equation 6.52) and using

the property that  $\sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) = 1$ , we get,

$$L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x} | Q_{\lambda}, Q_{\mathbf{z}}) = \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} C_{Q_{\mathbf{z},i}} - \int d\lambda Q_{\lambda}(\lambda) \ln \frac{Q_{\lambda}(\lambda)}{P(\lambda)}. \quad (6.58)$$

Substituting the expressions for  $P(\lambda)$  from Equation 6.13 and  $Q_{\lambda}(\lambda)$  from Equation 6.45, we obtain the following expression for the VB lower bound,

$$\begin{aligned} L_{\mathbf{Z}}^{\text{VB}} &= \sum_{i=1}^{|\mathcal{V}_{\mathbf{x}}|} C_{Q_{\mathbf{z},i}} - \text{KL}(\boldsymbol{\nu}^* \| \boldsymbol{\nu}^{\text{prior}}) - \sum_{z^s \in \mathcal{Z}^s} \text{KL}(\boldsymbol{\alpha}_{z^s}^* \| \boldsymbol{\alpha}_{z^s}^{\text{prior}}) \\ &\quad - \sum_{z^s \in \mathcal{Z}^s} \text{KL}(\boldsymbol{\beta}_{z^s}^{s*} \| \boldsymbol{\beta}_{z^s}^{s \text{prior}}) - \sum_{z^e \in \mathcal{Z}^e} \text{KL}(\boldsymbol{\beta}_{z^e}^{e*} \| \boldsymbol{\beta}_{z^e}^{e \text{prior}}), \end{aligned} \quad (6.59)$$

where,

$$\text{KL}(\boldsymbol{\nu}^* \| \boldsymbol{\nu}^{\text{prior}}) = \ln \frac{\Gamma(\boldsymbol{\nu}_0^*)}{\Gamma(\boldsymbol{\nu}_0^{\text{prior}})} - \sum_{j=1}^{\text{len}(\boldsymbol{\nu})} \left[ \ln \frac{\Gamma(\boldsymbol{\nu}_j^*)}{\Gamma(\boldsymbol{\nu}_j^{\text{prior}})} - (\boldsymbol{\nu}_j^* - \boldsymbol{\nu}_j^{\text{prior}}) (\psi(\boldsymbol{\nu}_j^*) - \psi(\boldsymbol{\nu}_0^*)) \right], \quad (6.60)$$

with,

$$\boldsymbol{\nu}_0 = \sum_{j=1}^{\text{len}(\boldsymbol{\nu})} \boldsymbol{\nu}_j. \quad (6.61)$$

The complete VBEM algorithm is summarized in Algorithm 6.2. The inputs to the algorithm are the training set  $\mathbf{x}$  that contain start/end handshake labels for signs in a vocabulary,  $\mathcal{V}_{\mathbf{x}}$  and two sets of Dirichlet distribution parameters: the initial values for the hyper-parameters,  $\boldsymbol{\omega}^{\circ}$ , and, the parameters for prior distributions over model parameters,  $\boldsymbol{\omega}^{\text{prior}}$ . The outputs of the VBEM algorithm are the estimated values for model parameters,  $\boldsymbol{\lambda}_{\mathbf{Z}}^{\text{VB}}$ .

The sequence of steps in the VBEM algorithm closely parallel those of the MAPEM algorithm. The M-step in the VBEM algorithm differs slightly from the MAPEM algorithm in that VBEM involves updating,  $\boldsymbol{\omega}$ , the hyper-parameters for Dirichlet distributions over model parameters while MAPEM involves directly updating the model parameters,  $\boldsymbol{\lambda}$ . The hyper-parameters therefore do not need to be normalized in each EM iteration. The E-step in both cases involves updating the variational distributions  $Q_{\mathbf{z},i}$  associated with each item  $i$  in the vocabulary and also the respective normalizing constants to be used in updating the

estimated lower bounds. After convergence of the algorithm, the model parameters,  $\lambda_Z^{\text{VB}}$ , are computed as the expected values of the Dirichlet distributions with parameters  $\omega^*$ .

### 6.3 Learning the HSBN<sup>congruent</sup> model parameters

In `two-handed:same handshapes` signs, the handshapes articulated on the two hands are the same, or, are very similar. The other properties pertaining to the change in handshape between the start and end points as well as the patterns of variation in handshape configuration are similar to that of `one-handed` signs. The learning formulations developed for `one-handed` signs can therefore be extended in a straightforward fashion to the HSBN<sup>congruent</sup> model. The corresponding MAPEM<sup>congruent</sup> and VBEM<sup>congruent</sup> formulations essentially involve a summation over the handshapes on the two hands. We present the corresponding equations below in the interest of completeness.

We will use a training set  $\mathbf{x}$  containing start/end handshape annotations for the productions of `two-handed:same handshapes` signs from a vocabulary,  $\mathcal{V}_x$ . The training set is arranged as,  $\mathbf{x} = \{\mathbf{x}_i\}$ , where  $i$  ranges over the items in the vocabulary. The productions of the  $i^{\text{th}}$  vocabulary item are denoted as,  $\mathbf{x}_i = \{\mathbf{x}_{ij}\}$ . The start/end handshapes annotated for each example are denoted using the tuple,  $\mathbf{x}_{ij} = (x_{ij}^{s;\text{D}}, x_{ij}^{s;\text{N}}, x_{ij}^{e;\text{D}}, x_{ij}^{e;\text{N}})$ . The label set for handshapes  $x$  is denoted as  $\mathcal{X}$ .

As in the case of the learning formulation for `one-handed` signs, each vocabulary item  $i$  is associated with a pair of hidden variables denoted as  $\mathbf{z}_i = (z_i^s, z_i^e)$ . The set of all hidden variables is denoted as  $\mathbf{z} = \{\mathbf{z}_i\}$ . The start and end latent variables take values from their associated state-spaces,  $\mathcal{Z}^s, \mathcal{Z}^e$ .

The expressions for the MAPEM and VBEM lower bounds remain the same as in the `one-handed` case, Equations 6.11 and 6.37.

The complete data log-likelihood term,  $\ln P(\mathbf{x}_i, z_i^s, z_i^e | \lambda)$ , in Equations 6.10 and 6.36 can be expanded given the plate representation for the training set depicted in Figure 6-2

**Algorithm 6.2:** VBEM<sup>dominant</sup> algorithm for learning HSBN<sup>dominant</sup> parameters

**Inputs** :  $\mathbf{x}$  Handshape label pairs for signs contained in a training set,  
:  $\omega^{\text{prior}}$  Prior distributions for HSBN parameters,  $\lambda$ ,  
:  $\omega^\circ$  Initial parameters of Dirichlet distributions for  $\lambda$ .

**Outputs**:  $\lambda_Z^{\text{VB}}$  Estimated HSBN parameters,  
:  $L_Z^{\text{VB}}$  Estimated VB lower bound to  $[\ln P(\mathbf{x})]$ .

- 1  $\omega^* \leftarrow \omega^\circ$ ;
- 2 **repeat**
  - /\* E-Step (derived in Equation 6.56) \*/
  - 3 **for**  $i \leftarrow 1$  to  $|\mathcal{V}_x|$  **do**
    - 4 
$$\tilde{Q}_{\mathbf{z},i}[z^s, z^e] \leftarrow \exp \left[ \psi(\nu^*[z^s]) - \psi \left( \sum_{\tilde{z}^s \in \mathcal{Z}^s} \nu^*[\tilde{z}^s] \right) + \psi(\alpha^*[z^s, z^e]) - \psi \left( \sum_{\tilde{z}^e \in \mathcal{Z}^e} \alpha^*[\tilde{z}^e, z^e] \right) \right. \\ \left. + \sum_{j=1}^{|\mathbf{x}_i|} \left[ \psi(\beta^{s*}[z^s, x_{ij}^s]) - \psi \left( \sum_{x \in \mathcal{X}} \beta^{s*}[z^s, x] \right) + \psi(\beta^{e*}[z^e, x_{ij}^e]) - \psi \left( \sum_{x \in \mathcal{X}} \beta^{e*}[z^e, x] \right) \right] \right]$$

$$\forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$$
  - 5 
$$C_{Q_z}[i] \leftarrow \sum_{z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e} \tilde{Q}_{\mathbf{z},i}[z^s, z^e]; \quad Q_{\mathbf{z},i}[z^s, z^e] \leftarrow \frac{\tilde{Q}_{\mathbf{z},i}[z^s, z^e]}{C_{Q_z}[i]} \quad \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$$
- end**
  - /\* Update the VBEM lower bound (derived in Equation 6.59) \*/
  - 6 
$$L_Z^{\text{VB}} \leftarrow \sum_{i=1}^{|\mathcal{V}_x|} \ln C_{Q_z}[i] - \text{KL}(\nu^* \| \nu^{\text{prior}}) - \sum_{z^s \in \mathcal{Z}^s} \text{KL}(\alpha^*[z^s, *] \| \alpha^{\text{prior}}[z^s, *]) \\ - \sum_{z^s \in \mathcal{Z}^s} \text{KL}(\beta^{s*}[z^s, *] \| \beta^{s \text{prior}}[z^s, *]) - \sum_{z^e \in \mathcal{Z}^e} \text{KL}(\beta^{e*}[z^e, *] \| \beta^{e \text{prior}}[z^e, *]);$$
  - /\* M-step (derived in Equation 6.46) \*/
  - 7 
$$\nu^*[z^s] \leftarrow \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}[z^s, z^e] + \nu^{\text{prior}}[z^s] \right] \quad \forall z^s \in \mathcal{Z}^s;$$
  - 8 
$$\alpha^*[z^s, z^e] \leftarrow \left[ \sum_{i=1}^{|\mathcal{V}_x|} Q_{\mathbf{z},i}[z^s, z^e] + \alpha^{\text{prior}}[z^s, z^e] \right] \quad \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$$
  - 9 
$$\beta^{s*}[z^s, x] \leftarrow \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}[z^s, z^e] \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^s) + \beta^{s \text{prior}}[z^s, x] \right] \quad \forall z^s \in \mathcal{Z}^s, x \in \mathcal{X};$$
  - 10 
$$\beta^{e*}[z^e, x] \leftarrow \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^s \in \mathcal{Z}^s} Q_{\mathbf{z},i}[z^s, z^e] \sum_{j=1}^{|\mathbf{x}_i|} \delta(x, x_{ij}^e) + \beta^{e \text{prior}}[z^e, x] \right] \quad \forall z^e \in \mathcal{Z}^e, x \in \mathcal{X};$$
- until** the lower bound,  $L_Z^{\text{VB}}$  and the parameters,  $\omega^*$  converge;
- 11  $\lambda_Z^{\text{VB}} \leftarrow \mathbb{E}_{Q_{\lambda}(\lambda | \omega^*)}[\lambda];$  /\* computed using Equation 6.48 \*/

as follows,

$$\begin{aligned} \ln P(\mathbf{x}_i, z_i^s, z_i^e | \boldsymbol{\lambda}) &= \ln \boldsymbol{\pi}_{z_i^s} + \ln \mathbf{a}_{z_i^s, z_i^e} \\ &+ \sum_{j=1}^{|\mathbf{x}_i|} \left[ \ln \mathbf{b}_{z_i^s}^s(x_{ij}^{s;D}) + \ln \mathbf{b}_{z_i^s}^s(x_{ij}^{s;N}) + \ln \mathbf{b}_{z_i^e}^e(x_{ij}^{e;D}) + \ln \mathbf{b}_{z_i^e}^e(x_{ij}^{e;N}) \right]. \end{aligned} \quad (6.62)$$

We present the ‘E’ and ‘M’ steps for the MAPEM<sup>congruent</sup> algorithm followed by the same for the VBEM<sup>congruent</sup> algorithm.

### 6.3.1 The MAPEM formulation for learning HSBN<sup>congruent</sup> model parameters

In the E-step the following updates for the hidden variable variational distributions,  $Q_{\mathbf{z}}(\mathbf{z})$ , are obtained using the sequence of steps as in Equations 6.16-6.19,

$$\begin{aligned} \ln Q_{\mathbf{z},i}(z^s, z^e) &= \ln \boldsymbol{\pi}_{z^s}^* + \ln \mathbf{a}_{z^s, z^e}^* \\ &+ \sum_{j=1}^{|\mathbf{x}_i|} \left[ \ln \mathbf{b}_{z^s}^{s*}(x_{ij}^{s;D}) + \ln \mathbf{b}_{z^s}^{s*}(x_{ij}^{s;N}) + \ln \mathbf{b}_{z^e}^{e*}(x_{ij}^{e;D}) + \ln \mathbf{b}_{z^e}^{e*}(x_{ij}^{e;N}) \right] \\ &- C_{Q_{\mathbf{z},i}}. \end{aligned} \quad (6.63)$$

For the M-step, the sequence of steps as in Equations 6.20-6.27 yields the desired updates for the model parameters,

$$\begin{aligned} \boldsymbol{\pi}_{z^s}^* &= \frac{1}{C_{\boldsymbol{\pi}}} \left[ \sum_{i=1}^{|\mathcal{Y}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s) + \left( \boldsymbol{\nu}_{z^s}^{\text{prior}} - 1 \right) \right], \\ \mathbf{a}_{z^s, z^e}^* &= \frac{1}{C_{\mathbf{a}_{z^s, z^e}}} \left[ \sum_{i=1}^{|\mathcal{Y}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s, z^e) + \left( \boldsymbol{\alpha}_{z^s, z^e}^{\text{prior}} - 1 \right) \right], \\ \mathbf{b}_{z^s}^{s*}(x) &= \frac{1}{C_{\mathbf{b}_{z^s}^s}} \left[ \sum_{i=1}^{|\mathcal{Y}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^s) \sum_{j=1}^{|\mathbf{x}_i|} \left[ \delta(x, x_{ij}^{s;D}) + \delta(x, x_{ij}^{s;N}) \right] + \left( \boldsymbol{\beta}_{z^s}^{\text{prior}}(x) - 1 \right) \right], \\ \mathbf{b}_{z^e}^{e*}(x) &= \frac{1}{C_{\mathbf{b}_{z^e}^e}} \left[ \sum_{i=1}^{|\mathcal{Y}_{\mathbf{x}}|} Q_{\mathbf{z},i}(z^e) \sum_{j=1}^{|\mathbf{x}_i|} \left[ \delta(x, x_{ij}^{e;D}) + \delta(x, x_{ij}^{e;N}) \right] + \left( \boldsymbol{\beta}_{z^e}^{\text{prior}}(x) - 1 \right) \right]. \end{aligned} \quad (6.64)$$

The expression for the update of the MAPEM lower bound remains the same as in the one-handed case, Equation 6.30. The complete MAPEM<sup>congruent</sup> algorithm is summarized in Algorithm 6.3.

### 6.3.2 The VBEM formulation for learning HSBN<sup>congruent</sup> model parameters

In the E-step, the following updates for the hidden variable variational distributions,  $Q_{\mathbf{z}}(\mathbf{z})$ , are obtained using the sequence of steps as in Equations 6.49-6.56,

$$\begin{aligned}
\ln Q_{\mathbf{z},i}(z^s, z^e) &= -C_{Q_{\mathbf{z},i}} \\
&+ \psi(\boldsymbol{\nu}_{z^s}^*) - \psi\left(\sum_{\tilde{z}^s \in \mathcal{Z}^s} \boldsymbol{\nu}_{\tilde{z}^s}^*\right) + \psi(\boldsymbol{\alpha}_{z^s, z^e}^*) - \psi\left(\sum_{\tilde{z}^e \in \mathcal{Z}^e} \boldsymbol{\alpha}_{z^s, \tilde{z}^e}^*\right) \\
&+ \sum_{j=1}^{|\mathbf{x}_i|} \left[ \psi\left(\boldsymbol{\beta}_{z^s}^{s*}(x_{ij}^{s;D})\right) + \psi\left(\boldsymbol{\beta}_{z^s}^{s*}(x_{ij}^{s;N})\right) - 2\psi\left(\sum_{x \in \mathcal{X}} \boldsymbol{\beta}_{z^s}^{s*}(x)\right) \right] \\
&+ \sum_{j=1}^{|\mathbf{x}_i|} \left[ \psi\left(\boldsymbol{\beta}_{z^e}^{e*}(x_{ij}^{e;D})\right) + \psi\left(\boldsymbol{\beta}_{z^e}^{e*}(x_{ij}^{e;N})\right) - 2\psi\left(\sum_{x \in \mathcal{X}} \boldsymbol{\beta}_{z^e}^{e*}(x)\right) \right],
\end{aligned} \tag{6.65}$$

where,  $\psi$  is the *digamma* function and  $C_{Q_{\mathbf{z},i}}$  are the normalizing constants for the variational distributions  $Q_{\mathbf{z},i}$ .

For the M-step, the updates for  $Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})$  approximating the desired posterior distributions  $P(\boldsymbol{\lambda}|\mathbf{x})$  are obtained following the same sequence of steps as in Equations 6.39-6.46. The expression for  $\ln Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})$  is the same as in Equation 6.45. The expressions for the Dirichlet

<b>Algorithm 6.3:</b> MAPEM <sup>congruent</sup> algorithm for learning HSBN <sup>congruent</sup> parameters	
<b>Inputs</b>	$\mathbf{x}$ Handshake labels for two-handed: same handshaped signs; : The other inputs, $\boldsymbol{\lambda}^\circ, \boldsymbol{\omega}^{\text{prior}}$ , are the same as in Algorithm 6.1.
<b>Outputs:</b>	$\boldsymbol{\lambda}_Z^{\text{MAP}}$ Estimated HSBN parameters.
1	$\boldsymbol{\lambda}^* \leftarrow \boldsymbol{\lambda}^\circ;$
2	<b>repeat</b>
	/* E-Step (derived in Equation 6.63) */
3	<b>for</b> $i \leftarrow 1$ to $ \mathcal{V}_x $ <b>do</b>
4	$\tilde{Q}_{\mathbf{z},i}[z^s, z^e] \leftarrow \exp \left[ \ln \boldsymbol{\pi}^*[z^s] + \ln \mathbf{a}^*[z^s, z^e] \right. \\ \left. + \sum_{j=1}^{ \mathbf{x}_i } \left( \ln \mathbf{b}^{s*}[z^s, x_{ij}^{s;\text{D}}] + \ln \mathbf{b}^{s*}[z^s, x_{ij}^{s;\text{N}}] + \ln \mathbf{b}^{e*}[z^e, x_{ij}^{e;\text{D}}] + \ln \mathbf{b}^{e*}[z^e, x_{ij}^{e;\text{N}}] \right) \right]$ $\forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$
5	Normalize the variational distributions as in Algorithm 6.1, step 5;
	<b>end</b>
	/* Update the MAPEM lower bound (derived in Equation 6.30) */
6	$L_Z^{\text{MAP}} \leftarrow$ same as in Algorithm 6.1, step 6;
	/* M-step (derived in Equation 6.64) */
7	$\boldsymbol{\pi}^*[z^s] \leftarrow \frac{1}{C_\pi} \left[ \sum_{i=1}^{ \mathcal{V}_x } \sum_{z^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}[z^s, z^e] + (\boldsymbol{\nu}^{\text{prior}}[z^s] - 1) \right] \quad \forall z^s \in \mathcal{Z}^s;$
8	$\mathbf{a}^*[z^s, z^e] \leftarrow \frac{1}{C_{\mathbf{a}_{z^s}}} \left[ \sum_{i=1}^{ \mathcal{V}_x } Q_{\mathbf{z},i}[z^s, z^e] + (\boldsymbol{\alpha}^{\text{prior}}[z^s, z^e] - 1) \right] \quad \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$
9	$\mathbf{b}^{s*}[z^s, x] \leftarrow \frac{1}{C_{\mathbf{b}_{z^s}^s}} \left[ \sum_{i=1}^{ \mathcal{V}_x } \sum_{z^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}[z^s, z^e] \sum_{j=1}^{ \mathbf{x}_i } \left[ \delta(x, x_{ij}^{s;\text{D}}) + \delta(x, x_{ij}^{s;\text{N}}) \right] + (\boldsymbol{\beta}^{s \text{ prior}}[z^s, x] - 1) \right]$ $\forall z^s \in \mathcal{Z}^s, x \in \mathcal{X};$
10	$\mathbf{b}^{e*}[z^e, x] \leftarrow \frac{1}{C_{\mathbf{b}_{z^e}^e}} \left[ \sum_{i=1}^{ \mathcal{V}_x } \sum_{z^s \in \mathcal{Z}^s} Q_{\mathbf{z},i}[z^s, z^e] \sum_{j=1}^{ \mathbf{x}_i } \left[ \delta(x, x_{ij}^{e;\text{D}}) + \delta(x, x_{ij}^{e;\text{N}}) \right] + (\boldsymbol{\beta}^{e \text{ prior}}[z^e, x] - 1) \right]$ $\forall z^e \in \mathcal{Z}^e, x \in \mathcal{X};$
	<b>until</b> the lower bound, $L_Z^{\text{MAP}}$ <b>and</b> the parameters, $\boldsymbol{\lambda}^*$ converge;
11	$\boldsymbol{\lambda}_Z^{\text{MAP}} \leftarrow \boldsymbol{\lambda}^*;$



parameters extend from the one-handed case (Equation 6.46) and are given below,

$$\begin{aligned}
\boldsymbol{\nu}_{z^s}^* &= \boldsymbol{\nu}_{z^s}^{\text{prior}} + \sum_{i=1}^{|\mathcal{V}_x|} Q_{\mathbf{z},i}(z^s), \\
\boldsymbol{\alpha}_{z^s,z^e}^* &= \boldsymbol{\alpha}_{z^s,z^e}^{\text{prior}} + \sum_{i=1}^{|\mathcal{V}_x|} Q_{\mathbf{z},i}(z^s, z^e), \\
\boldsymbol{\beta}_{z^s}^{s*}(x) &= \boldsymbol{\beta}_{z^s}^{s \text{ prior}}(x) + \sum_{i=1}^{|\mathcal{V}_x|} Q_{\mathbf{z},i}(z^s) \sum_{j=1}^{|\mathbf{x}_i|} \left[ \delta(x, x_{ij}^{s;\text{D}}) + \delta(x, x_{ij}^{s;\text{N}}) \right], \\
\boldsymbol{\beta}_{z^s}^{e*}(x) &= \boldsymbol{\beta}_{z^s}^{e \text{ prior}}(x) + \sum_{i=1}^{|\mathcal{V}_x|} Q_{\mathbf{z},i}(z^e) \sum_{j=1}^{|\mathbf{x}_i|} \left[ \delta(x, x_{ij}^{e;\text{D}}) + \delta(x, x_{ij}^{e;\text{N}}) \right]. \quad (6.66)
\end{aligned}$$

The expression to update the VB lower bound remains the same as in the one-handed case, Equation 6.59. The complete VBEM<sup>congruent</sup> algorithm is summarized in Algorithm 6.4.

The learning algorithms for one-handed signs can be obtained as a special case of the algorithms for two-handed:same handshapes signs by leaving out the terms that involve variables depicting handshapes on the non-dominant hand (the corresponding equations are Equations 6.63, 6.64 and Equations 6.65, 6.66). Therefore only the MAPEM<sup>congruent</sup> and VBEM<sup>congruent</sup> versions need to be implemented and these algorithms utilize both one-handed and two-handed:same handshapes signs contained in the training set.

## 6.4 Summary

Learning the HSBN model involves estimating a state-space for the hidden variables and the parameters for multinomial distributions contained in the model. In this chapter we developed learning formulations for the parameter estimation task assuming a training set containing examples of monomorphemic lexical signs in a vocabulary along with their associated start/end handshape labels. The learning formulations were first developed for one-handed signs and subsequently extended to also include two-handed signs. We considered the MAPEM and the VBEM learning formulations for HSBN model parameter estimation. These two approaches are briefly summarized in Tables 6.3 and 6.4. Despite their many similarities, the key difference between the two approaches lies in the objective function

**Algorithm 6.4:** VBEM<sup>congruent</sup> formulation for learning HSBN<sup>congruent</sup> parameters

**Inputs** :  $\mathbf{x}$  Handshape labels for two-handed : same handshapes signs;  
: The other inputs,  $\omega^{\text{prior}}, \omega^{\circ}$ , are the same as in Algorithm 6.2.

**Outputs**:  $\lambda_Z^{\text{VB}}$  Estimated HSBN parameters,  
:  $L_Z^{\text{VB}}$  Estimated VB lower bound to  $[\ln P(\mathbf{x})]$ .

- 1  $\omega^* \leftarrow \omega^{\circ};$
- 2 **repeat**
  - /\* E-Step (derived in Equation 6.65) \*/
  - 3 **for**  $i \leftarrow 1$  to  $|\mathcal{V}_x|$  **do**
    - 4 
$$\begin{aligned} \tilde{Q}_{\mathbf{z},i}[z^s, z^e] \leftarrow & \exp \left[ \psi(\nu^*[z^s]) - \psi \left( \sum_{\tilde{z}^s \in \mathcal{Z}^s} \nu^*[\tilde{z}^s] \right) \right. \\ & + \psi(\alpha^*[z^s, z^e]) - \psi \left( \sum_{\tilde{z}^e \in \mathcal{Z}^e} \alpha^*[\tilde{z}^s, \tilde{z}^e] \right) \\ & + \sum_{j=1}^{|\mathbf{x}_i|} \left[ \psi(\beta^{s*}[z^s, x_{ij}^{\text{D}}]) + \psi(\beta^{s*}[z^s, x_{ij}^{\text{N}}]) - 2\psi \left( \sum_{x \in \mathcal{X}} \beta^{s*}[z^s, x] \right) \right] \\ & \left. + \sum_{j=1}^{|\mathbf{x}_i|} \left[ \psi(\beta^{e*}[z^e, x_{ij}^{\text{D}}]) + \psi(\beta^{e*}[z^e, x_{ij}^{\text{N}}]) - 2\psi \left( \sum_{x \in \mathcal{X}} \beta^{e*}[z^e, x] \right) \right] \right] \\ & \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e; \end{aligned}$$
    - 5 Normalize the variational distributions as in Algorithm 6.2, step 5;
    - end**
    - /\* Update the VBEM lower bound (derived in Equation 6.59) \*/
    - 6  $L_Z^{\text{VB}} \leftarrow$  same as in Algorithm 6.2, step 6;
    - /\* M-step (derived in Equation 6.66) \*/
    - 7 
$$\nu^*[z^s] \leftarrow \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}[z^s, z^e] + \nu^{\text{prior}}[z^s] \right] \quad \forall z^s \in \mathcal{Z}^s;$$
    - 8 
$$\alpha^*[z^s, z^e] \leftarrow \left[ \sum_{i=1}^{|\mathcal{V}_x|} Q_{\mathbf{z},i}[z^s, z^e] + \alpha^{\text{prior}}[z^s, z^e] \right] \quad \forall z^s \in \mathcal{Z}^s, z^e \in \mathcal{Z}^e;$$
    - 9 
$$\beta^{s*}[z^s, x] \leftarrow \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}[z^s, z^e] \sum_{j=1}^{|\mathbf{x}_i|} \left[ \delta(x, x_{ij}^{\text{D}}) + \delta(x, x_{ij}^{\text{N}}) \right] + \beta^{s \text{ prior}}[z^s, x] \right]$$
  
 $\forall z^s \in \mathcal{Z}^s, x \in \mathcal{X};$
    - 10 
$$\beta^{e*}[z^e, x] \leftarrow \left[ \sum_{i=1}^{|\mathcal{V}_x|} \sum_{z^s \in \mathcal{Z}^s} Q_{\mathbf{z},i}[z^s, z^e] \sum_{j=1}^{|\mathbf{x}_i|} \left[ \delta(x, x_{ij}^{\text{D}}) + \delta(x, x_{ij}^{\text{N}}) \right] + \beta^{e \text{ prior}}[z^e, x] \right]$$
  
 $\forall z^e \in \mathcal{Z}^e, x \in \mathcal{X};$
  - until** the lower bound,  $L_Z^{\text{VB}}$  and the parameters,  $\omega^*$  converge;
  - 11  $\lambda_Z^{\text{VB}} \leftarrow \mathbb{E}_{Q_{\lambda}(\lambda|\omega^*)}[\lambda];$  /\* computed using Equation 6.48 \*/

Notation	Description
$\ln P(\boldsymbol{\lambda}   \mathbf{x})$	MAP objective: $\max_{\boldsymbol{\lambda}} [\ln P(\mathbf{x}   \boldsymbol{\lambda}) + \ln P(\boldsymbol{\lambda}   \boldsymbol{\omega}^{\text{prior}})]$
$L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda}   \mathbf{x}, Q_{\mathbf{z}})$	Lower bound to the posterior distribution $[\ln P(\boldsymbol{\lambda}   \mathbf{x})]$ maximized by the MAPEM formulation: $\max_{\boldsymbol{\lambda}, Q_{\mathbf{z}}} [L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda}   \mathbf{x}, Q_{\mathbf{z}})]$
$Q_{\mathbf{z}}$	Variational distributions, $Q_{\mathbf{z}}$ , for hidden variables $\mathbf{z}$ are introduced to formulate the lower bound $L_{\mathbf{Z}}^{\text{MAP}}$ : $Q_{\mathbf{z}}(\mathbf{z}) = \{Q_{\mathbf{z},i}(z_i^s, z_i^e)\}, 1 \leq i \leq  \mathcal{V}_{\mathbf{x}} ,$ $\sum_{z_i^s \in \mathcal{Z}^s, z_i^e \in \mathcal{Z}^e} Q_{\mathbf{z},i}(z_i^s, z_i^e) = 1, \quad Q_{\mathbf{z},i}(z_i^s, z_i^e) \geq 0$
MAPEM E-step	$\max_{Q_{\mathbf{z}}} [L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda}   \mathbf{x}, Q_{\mathbf{z}})]$ yields an update for $Q_{\mathbf{z}}(\mathbf{z})$
MAPEM M-step	$\max_{\boldsymbol{\lambda}} [L_{\mathbf{Z}}^{\text{MAP}}(\boldsymbol{\lambda}   \mathbf{x}, Q_{\mathbf{z}})]$ yields an update for $\boldsymbol{\lambda}^*$
$\boldsymbol{\lambda}_{\mathbf{Z}}^{\text{MAP}}$	MAPEM estimated parameters for the HSBN model

**Table 6.3:** Summary of the MAPEM formulation for learning the HSBN.

Notation	Description
$\ln P(\mathbf{x})$	Complete data log-likelihood for the training set $\mathbf{x}$
$L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x}   Q_{\boldsymbol{\lambda}}, Q_{\mathbf{z}})$	Lower bound for data log-likelihood $[\ln P(\mathbf{x})]$ maximized by the VBEM formulation: $\max_{Q_{\boldsymbol{\lambda}}, Q_{\mathbf{z}}} [L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x}   Q_{\boldsymbol{\lambda}}, Q_{\mathbf{z}})]$
$Q_{\mathbf{z}}$	Variational distributions, $Q_{\mathbf{z}}$ , for hidden variables $\mathbf{z}$ . These are defined the same as in the MAPEM formulation above
$Q_{\boldsymbol{\lambda}}$	Variational distributions, $Q_{\boldsymbol{\lambda}}$ , for model parameters $\boldsymbol{\lambda}$ are introduced to derive the lower bound $L_{\mathbf{Z}}^{\text{VB}}$ ; obtained as: $\ln Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}   \boldsymbol{\omega}^*) = \ln \text{Dir}(\boldsymbol{\pi}   \boldsymbol{\nu}^*) + \sum_{z^s} \ln \text{Dir}(\mathbf{a}_{z^s, \cdot}   \boldsymbol{\alpha}_{z^s, \cdot}^*)$ $+ \sum_{z^s} \ln \text{Dir}(\mathbf{b}_{z^s}^{s*}(\cdot)   \boldsymbol{\beta}_{z^s}^{s*}(\cdot)) + \sum_{z^e} \ln \text{Dir}(\mathbf{b}_{z^e}^{e*}(\cdot)   \boldsymbol{\beta}_{z^e}^{e*}(\cdot))$
VBEM E-step	$\max_{Q_{\mathbf{z}}} [L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x}   Q_{\boldsymbol{\lambda}}, Q_{\mathbf{z}})]$ yields an update for $Q_{\mathbf{z}}(\mathbf{z})$
VBEM M-step	$\max_{Q_{\boldsymbol{\lambda}}} [L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x}   Q_{\boldsymbol{\lambda}}, Q_{\mathbf{z}})]$ yields an update for $\boldsymbol{\omega}^*$
$\boldsymbol{\lambda}_{\mathbf{Z}}^{\text{VB}} = \mathbb{E}_{Q_{\boldsymbol{\lambda}}(\boldsymbol{\lambda}   \boldsymbol{\omega}^*)} [\boldsymbol{\lambda}]$	VBEM estimated parameters for the HSBN model

**Table 6.4:** Summary of the VBEM formulation for learning the HSBN.

chosen for optimization. Because the VBEM approach aims to estimate the total data log-likelihood for a given training set, it requires an integration over the space of model parameters. The VBEM objective function therefore encapsulates an implicit penalty for model complexity [Beal, 2003] – a property that will prove instrumental for the hidden variable state-space estimation approach formulated in the next chapter.

An arrangement of the training set into groups of different productions of signs with each group associated with one specific lexical item in the vocabulary allows the learning algorithm to accrue patterns of sign-independent handshape variation. Because each lexical item is associated with one pair of hidden variable states, one-to-many associations are produced between the hidden (unobserved) variables and the handshape labels observed (annotated) in the training set. This property constitutes the key difference between the learning formulations developed here for parameter estimation in the HSNB vis-a-vis the learning formulations for parameter estimation in the HMM [Beal, 2003].

As with other Expectation Maximization approaches, the HSNB parameter estimation algorithms are gradient ascent based and therefore the convergence to a local optimum is guaranteed. However, the algorithms are sensitive to initialization. We describe one approach to perform model initialization in the next chapter. The other aspects of the HSNB parameter estimation are straightforward to implement and are computationally efficient.

## Chapter 7

# Learning a State-space for Hidden Variables in the HSBN

In this chapter we formulate the HSBNStateSpaceEstimation algorithm to learn a suitable state-space,  $\hat{\mathbf{Z}} = (\hat{\mathbf{Z}}^s, \hat{\mathbf{Z}}^e)$ , to represent hidden variables,  $(z^s, z^e)$ , in the HSBN.

In a reference implementation we may assume that the cardinality of the set of hidden states corresponds to the cardinality of the set of observed handshape labels, i.e.,  $\hat{\mathbf{Z}} := (\mathcal{X}, \mathcal{X})$ . The respective model parameters,  $\lambda_{\hat{\mathbf{Z}}}$ , are then estimated given a training set,  $\mathbf{x}$ , using either the MAPEM or VBEM algorithms presented in the previous chapter. This reference implementation suffers from the drawback that it involves a large number of hidden states and therefore requires a commensurate number of free parameters to be estimated during the learning. As a consequence, the learnt model can more easily accrue statistical irregularities contained in the training set (this is especially the case when the model is trained using datasets with a modest number of examples as are currently available for sign language research). The ability of the learnt model to generalize to unseen data is crucial for robust performance in a person-independent recognition task. This aspect therefore motivates the question of whether a different state-space representation,  $\tilde{\mathbf{Z}}$ , presumably with a smaller number of hidden states, could be inferred given the training set towards improving the generalization performance of the estimated model (with corresponding parameters  $\lambda_{\tilde{\mathbf{Z}}}$ ).

Optimization based learning approaches for estimating the hidden variable state-space aim to reduce the structural complexity of the model learnt given a training set. This is typically accomplished in two ways. A regularization term that consists of priors for the probability distribution parameters contained in the model (often also referred to as

smoothness priors) is included in the learning objective. Additionally, the objective function incorporates a bias towards models with a smaller number of free parameters. This preference can either be included as an explicit term in the objective function (e.g., BIC uses a function of the number of free parameters in a model as a measure of its structural complexity) or can arise as an implicit property of the objective function formulated for learning (e.g., through an integration performed over the space of model parameters in the variational Bayes approach [Beal, 2003]). A learning algorithm is subsequently formulated to optimize the objective function chosen for state-space estimation. A combinatorial optimization approach is necessary when a collection of discrete states are used to represent hidden variables in the model (as is the case for the HSBN).

In this chapter we investigate a stochastic optimization approach to estimate the HSBN hidden variable state-space<sup>1</sup>. We select the variational Bayes lower bound estimated by the VBEM approach as the optimization objective to maximize via the state-space search algorithm. The motivations behind this choice are described in the next section. The state-space estimation algorithm employs a sequence of learning ‘epochs’ to iteratively refine the state-space with a goals towards increasing the estimated VB lower bound. The algorithm can either start with a small number of randomly specified hidden states and augment this set with new states in the subsequent learning epochs, or alternatively, can start with a large number of explicitly initialized hidden states and attempt to reduce this number in the subsequent epochs. We use the latter approach for the HSBN since, as in the reference implementation, the set of handshape labels determined by linguists for annotating signs serves as an appropriate initial representation for the hidden variable states. The algorithm generates candidates for the state-space in the next epoch by applying one of the following methods for state-space refinement: {merge-states, drop-state, reset-state, add-state}. The selection of a state-space from among these candidates is based on the degree to which the generated hypothesis improves the VB lower bound. The algorithm stops when no further improvement to the VB lower bound is possible. The state-space estimation strat-

---

<sup>1</sup>Other optimization formulations such as reversible jump Markov chain Monte-Carlo (RJMCMC) [Green, 1995] would also be applicable in this context.

egy adopted here is in essence a local search formulation wherein the model initialized in the first learning epoch serves to anchor the state-space exploration. The complete `HSBNStateSpaceEstimation` algorithm is summarized in Algorithm 7.1. The different steps of this algorithm are described in more detail in the rest of this chapter.

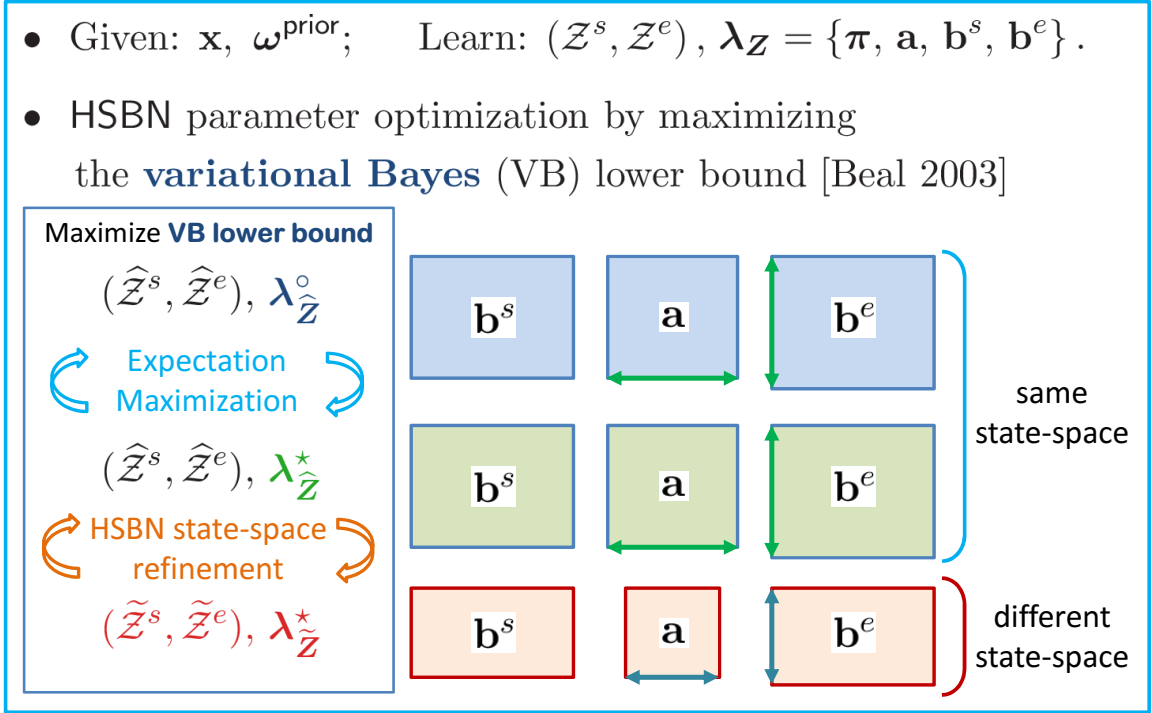
## 7.1 Overview of the `HSBNStateSpaceEstimation` algorithm

The `HSBNStateSpaceEstimation` algorithm for learning the HSBN is developed as follows. We utilize the variational Bayes lower bound to the complete data log-likelihood,  $\ln P(\mathbf{x})$ , specifically, the lower bound  $L_{\mathbf{Z}}^{\text{VB}}$  estimated by the VBEM algorithm, as the learning objective to maximize with respect to the hidden variable states  $\mathbf{Z}$ ,

$$\left[ \hat{\mathbf{Z}}, \omega_{\hat{\mathbf{Z}}} \right] \leftarrow \arg \max_{\mathbf{Z}, \omega_{\mathbf{Z}}} \left[ L_{\mathbf{Z}}^{\text{VB}}(\mathbf{x}) \right]. \quad (7.1)$$

In formulating this lower bound (Equations 6.31 - 6.37), the VB approach incorporates Dirichlet priors for multinomial parameters contained in the model. The VB lower bound also incorporates a bias towards models with a smaller number of free parameters because of the integration over model parameters performed in Equation 6.32. A computationally attractive feature of the VBEM formulation in the HSBN context is that it yields closed form expressions for the different steps and therefore circumvents the implementation complexity of sampling based formulations that can also provide other lower bounds to the complete data log-likelihood. The general theme of utilizing the VB lower bound for the purpose of comparing among models with different complexities was employed for estimating Gaussian mixtures with an unknown number of mixture components in [Beal, 2003]. A somewhat different combinatorial approach is required for the HSBN because its state-space is discrete.

An overview of the optimization formulation for learning the HSBN is illustrated in Figure 7.1. Given a training set,  $\mathbf{x}$ , and, the parameters of Dirichlet priors for model parameters,  $\omega^{\text{prior}}$ , the `HSBNStateSpaceEstimation` algorithm iteratively adapts the state-space for hidden variables,  $\mathbf{Z}_{\tau}$ , in a sequence of learning epochs (the epochs are indexed by  $\tau$ ) with a goal towards increasing the value of the estimated VBEM lower bound,  $L_{\tau}^{\text{VB}}$ . In each epoch,



**Figure 7-1:** An overview of the proposed optimization formulation for learning the HSNB parameters.

the algorithm selects from among a set of state-space hypotheses to determine an appropriate state-space candidate to use in the next epoch. The hyper-parameters estimated in an epoch, after state-space refinement, serve as the initialization for state-space hypotheses in the next epoch, i.e.,  $\omega_{\tau}^* \rightsquigarrow \{\omega_{\tau+1,k}^{\circ}\}$ . An alternate (essentially equivalent) implementation that we do not consider here would be to apply the state-space refinements to the hidden variable variational distributions, i.e.,  $Q_{\mathbf{z},\tau}^* \rightsquigarrow \{Q_{\mathbf{z},\tau+1,k}^{\circ}\}$ . To aid with circumventing local maxima in the objective function, the selection of a state-space from among the set of hypotheses is performed in a stochastic fashion. Larger improvements in the VB lower bound are assigned higher probabilities for being selected than hypotheses with smaller improvements (or even negative changes) in the estimated value of the VBEM lower bound.



## 7.2 Initializing the HSBN state-space

The initialization of the hidden variable states,  $\mathbf{Z}_{\tau=1}$ , is performed explicitly in the first learning epoch by providing the initial values for the variational distributions,  $Q_{\mathbf{z}}^{\circ} = \{Q_{\mathbf{z},i}^{\circ}(z^s, z^e)\}$ . These variational distributions are then used as the initial conditions for the VBEM algorithm to estimate the hyper-parameters in the first epoch.

The HSBNStateSpaceInitialization algorithm to initialize the HSBN learning is summarized in Algorithm 7.2. The set of handshake labels,  $\mathcal{X}$ , are utilized to serve as an initial representation,  $\mathbf{Z}_{\tau=1}$ , for the hidden variable states in Algorithm 7.2, step 2. The algorithm then aggregates the start/end handshake pairs annotated in the training set for the examples of each vocabulary item  $i$  into the respective initial variational distributions,  $Q_{\mathbf{z},i}^{\circ}$  in Algorithm 7.2, steps 4-8. These distributions are subsequently used in Algorithm 7.2, steps 9-16 to initialize the VBEM algorithm towards estimating the hyper-parameters,  $\omega_{\tau=1}$ , for the first epoch.

## 7.3 Hyper-parameters for the prior distributions

Hyper-parameters,  $\omega^{\text{prior}}$ , for prior distributions over model parameters play an important role in the VBEM algorithm. The priors influence both the estimated variational Bayes lower bound and the estimated model parameters. Uniform priors are frequently chosen to serve as the regularization terms during parameter estimation. In some cases, it is feasible to construct informative priors that reflect certain underlying properties that are relevant to the problem domain. For the HSBN, we construct Dirichlet priors in the first learning epoch with hyper-parameters,  $\omega_{\tau=1}^{\text{prior}} = \{\nu_{\tau=1}^{\text{prior}}, \alpha_{\tau=1}^{\text{prior}}, \beta_{\tau=1}^s, \beta_{\tau=1}^e\}$ . The procedure used for specifying the hyper-parameters for the prior distributions will be described in the experiments chapter.

In the HSBNStateSpaceEstimation algorithm, the prior distributions specified in the first epoch,  $\omega_{\tau=1}^{\text{prior}}$ , are propagated forwards through the subsequent epochs by applying the same state-space refinement methods as those employed for the hyper-parameters,  $\omega_{\tau}$ . The only difference being that a max operation is employed instead of a summation in

the merge-states state-space refinement (the corresponding lines in the pseudo-code listing are Algorithm 7.4, steps 7-9, Algorithm 7.4, steps 15-16). The hyper-parameters for priors are omitted from the following presentation in the interest of clarity.

#### 7.4 HSBN state-space refinement

In each learning epoch, the `HSBNStateSpaceEstimation` algorithm generates state-space candidates for the next epoch,  $\{\mathbf{Z}_{\tau+1,k}\}$ , by applying different state-space refinement methods to modify the current state-space,  $\mathbf{Z}_\tau$ . The methods chosen in our implementation are denoted as  $\{\text{MergeHSBNstates}, \text{DropHSBNstate}, \text{ResetHSBNstate}, \text{AddHSBNstate}\}$ . These methods correspond to accumulating the properties of a pair of hidden variable states into a single state, to disregarding the properties of a selected state in order that the remaining states can adopt its properties, to resetting the properties of a selected state to the corresponding values estimated during model initialization and to augmenting the set of hidden states with an additional state. The reset-state and add-state refinements were specifically chosen to allow the `HSBNStateSpaceEstimation` algorithm to revert changes that were performed to the state-space during drop-state and merge-state refinements in earlier epochs.

The state-space candidates generated in the epoch  $\tau$  are denoted as  $\{\mathbf{Z}_{\tau+1,k}\} = \{\mathbf{Z}_\tau^{\text{merge}:\rho,l\leftarrow m}\} \cup \{\mathbf{Z}_\tau^{\text{drop}:\rho,n}\} \cup \{\mathbf{Z}_\tau^{\text{reset}:\rho,o}\} \cup \{\mathbf{Z}_\tau^{\text{add}:\rho,p}\}$ . The subscript  $k$  in the LHS indexes the items in the generated set. The superscript  $\rho$  denotes whether the state-space refinement is applied to the start or the end hidden variable state-space (i.e, either  $\mathbf{Z}_\tau^s$  and  $\mathbf{Z}_\tau^e$ ). As the state-space  $\mathbf{Z}_\tau$  evolves during the learning, each of the state-space refinement algorithms are designed so as to retain the associations of hidden variable states with the states chosen for initialization in the first epoch. Ensuring this property simplifies the formulation of the proposed state-space refinement algorithms. The superscripts  $l, m, n, o, p$  are therefore indices into the initial set of hidden variable states,  $\mathbf{Z}_{\tau=1}$ .

The state-space candidates in an epoch of the `HSBNStateSpaceEstimation` algorithm

are produced by means of applying the above mentioned state-space refinements to the hyper-parameters associated with the current epoch,  $\omega_\tau$ . The merge-states, drop-state and reset-state refinements are straightforward to perform. We adopt a simple approach for the add-state refinement wherein the properties of the hyper-parameters computed in the first epoch for a selected state are included into the current set of hyper-parameters. The generated hyper-parameters and their associated VBEM lower bounds are denoted as,  $\{\omega_{\tau+1,k}\}, \{L_{\tau+1,k}^{\text{VB}}\}$ .

The different state-space refinements are formulated as follows.

- **MergeHSBNstates**

The algorithm for the MergeHSBNstates state-space refinement is summarized in Algorithm 7.4. The inputs are the training set  $\mathbf{x}$ , the hyper-parameters for the current epoch  $\omega_\tau$ , whether a start or an end state has been chosen for applying the merge state-space refinement  $\rho$ , and, the indices for the pair of states selected to be merged  $\psi, \varphi$ . The following are obtained as outputs after applying the specified state-space refinement: the hyper-parameters,  $\omega_\tau^{\text{merge}:\rho,\psi\leftarrow\varphi}$ ; the VBEM lower bound,  $L_\tau^{\text{merge}:\rho,\psi\leftarrow\varphi}$ ; and the state-space,  $\mathbf{Z}_\tau^{\text{merge}:\rho,\psi\leftarrow\varphi}$ . The candidate hyper-parameters,  $\tilde{\omega}$ , are obtained by summing together either a pair of rows or columns, indexed as specified by the inputs  $(\rho, \psi, \varphi)$ , in the hyper-parameters for the current epoch,  $\omega_\tau$ . The steps used to compute the candidate hyper-parameters are listed in Algorithm 7.4, steps 1 - 16. For the purposes of retaining the hidden variable state associations through the learning epochs, the state  $\varphi$  is regarded as having been incorporated into the state  $\psi$ . The candidate hyper-parameters,  $\tilde{\omega}$ , are used as initialization for the VBEM algorithm in Algorithm 7.4, step 17 in order to compute the transformed hyper-parameters,  $\omega_\tau^{\text{merge}:\rho,\psi\leftarrow\varphi}$ , and the associated VB lower bound,  $L_\tau^{\text{merge}:\rho,\psi\leftarrow\varphi}$ . In a subsequent learning epoch, the ResetHSBNstate and AddHSBNstate methods compute  $\omega_\tau^{\text{reset}:\rho,\psi}$ ,  $\omega_\tau^{\text{add}:\rho,\varphi}$  that serve to revert the changes performed by the above merge state-space refinement to the target and source states,  $\psi$  and  $\varphi$ .

- **DropHSBNstate**

The algorithm for the DropHSBNstate state-space refinement is summarized in Algorithm 7.5. The inputs provided are similar to that of the MergeHSBNstates algorithm and include  $\rho$  that denotes whether a start or an end state has been chosen for the drop-state state-space refinement, and, the index  $\psi$  of the state selected to be dropped from the current state-space. The outputs are same as those of the MergeHSBNstates algorithm. The candidate hyper-parameters,  $\tilde{\omega}$ , are obtained by removing either a row or a column, indexed as specified in the inputs  $(\rho, \psi)$ . The steps used to compute the candidate hyper-parameters are listed in Algorithm 7.5, steps 1 - 11. The updated set of hyper-parameters,  $\omega_{\tau}^{\text{drop}:\rho, \psi}$ , and the VB lower bound,  $L_{\tau}^{\text{drop}:\rho, \psi}$  are obtained as listed in Algorithm 7.5, step 12.

The AddHSBNstate method serves the role of reverting a DropHSBNstate operation performed in a previous epoch.

- **ResetHSBNstate**

The algorithm for the ResetHSBNstate state-space refinement is summarized in Algorithm 7.6. The inputs  $\rho, \psi$  provided are the same as that of the DropHSBNstate algorithm. The candidate hyper-parameters,  $\tilde{\omega}$ , are obtained by replacing the values of either a row or a column (indexed as specified in the inputs  $\rho, \psi$ ) with the corresponding values of the hyper-parameters,  $\omega_{\tau=1}$ , estimated in the first learning epoch. The steps used to compute the candidate hyper-parameters are listed in Algorithm 7.6, steps 5 - 10. The updated set of hyper-parameters,  $\omega_{\tau}^{\text{reset}:\rho, \psi}$ , and the VB lower bound,  $L_{\tau}^{\text{reset}:\rho, \psi}$  are obtained as listed in Algorithm 7.6, step 11.

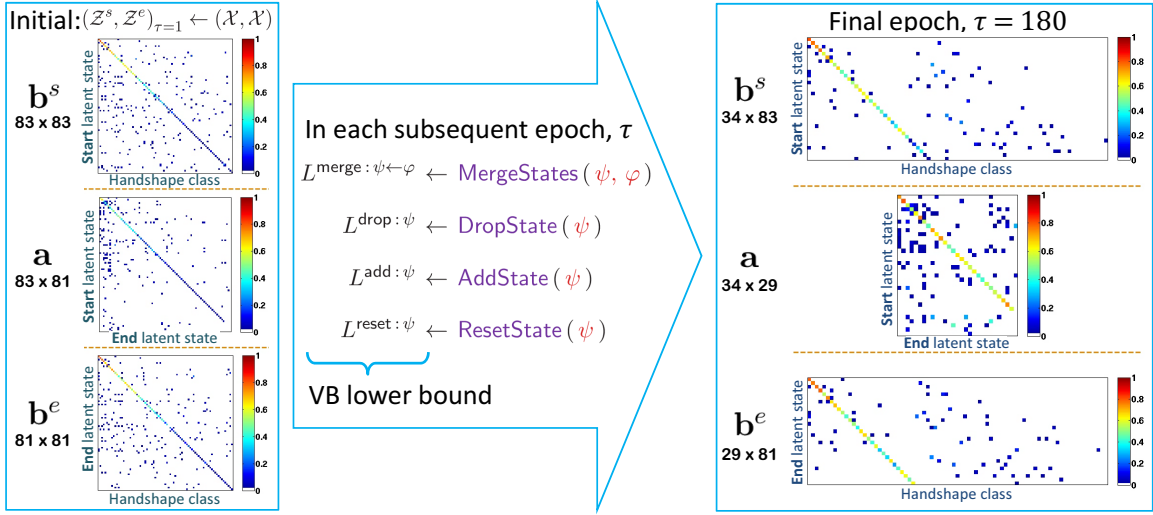
- **AddHSBNstate**

The algorithm for the AddHSBNstate state-space refinement is summarized in Algorithm 7.7. The hidden variable state  $\psi$  specified in the inputs for the purposes of incorporating into the current state-space belongs to the initial set of hidden variable states,  $\mathcal{Z}_{\tau=1}^{\rho}$ . The AddHSBNstate state-space refinement is accomplished by incor-

porating the values of hyper-parameters estimated in the first epoch  $\omega_{\tau=1}$  for the selected hidden variable state indexed by  $(\rho, \psi)$  into the hyper-parameters for the current epoch,  $\omega_{\tau}$ . These steps are listed in Algorithm 7.7, steps 5-12. The updated set of hyper-parameters,  $\omega_{\tau}^{\text{add}:\rho, \psi}$ , and the VB lower bound,  $L_{\tau}^{\text{add}:\rho, \psi}$ , are obtained as listed in Algorithm 7.7, step 13.

The HSBNStateSpaceSelection algorithm to generate a collection of state-space candidates by applying different state-space refinements to the current (epoch  $\tau$ ) state-space and to subsequently select a candidate state-space for the next learning epoch ( $\tau + 1$ ) is summarized in Algorithm 7.3. The state-space and hyper-parameters for the current epoch,  $\tau$ , (and also for the first epoch,  $\tau = 1$ ) are provided as inputs. The outputs produced are parameters for the next epoch. The four different state-space refinements (merge, drop, reset, add) are applied to the current state-space to produce a list of state-space candidates, Algorithm 7.3, steps 1-8. Applying each of these state-space refinements involves iterating over the different possible state selections and computing the updated parameters and VB lower bounds. Specifically, the merge-state state-space refinement iterates over pairs of states, the drop-state and reset-state refinements iterate over individual states and the add-state state-space refinement iterates over the states in the initial state-space not contained in the current state-space. Given the VB lower bounds computed for each of the generated state-space candidates, their difference with the current VB lower bound is taken, Algorithm 7.3, step 9 (we may recall here that the lower bound computed in the VBEM algorithm is the log of the true value). An approximately equal number of candidates for the different refinement types are chosen from among the generated candidates. The number of latter candidates selected are an implementation choice. The differences in VB lower bounds are transformed into an acceptance ratio based sampling distribution, Algorithm 7.3, step 11. A state-space candidate for the next epoch is then sampled from this distribution, Algorithm 7.3, steps 12-13.

The HSBNStateSpaceEstimation algorithm summarized in Algorithm 7.1 brings together all the different aspects of the HSBN learning formulation. The HSBNStateSpaceInitializa-



**Figure 7-2:** An illustration of results produced using the proposed algorithm for learning the HSBN. Parameters obtained after model initialization are displayed in the left column. The state-space refinement methods employed to generate model candidates in each epoch are listed in the center column. The estimated start/end latent states and model parameters in the final epoch after convergence of the variational Bayes lower bound are displayed in the last column.

tion algorithm is used to initialize the state-space in Algorithm 7.1, step 1 and the HSBN-StateSpaceSelection algorithm is used to iteratively refine the state-space towards increasing the estimated VB lower bound Algorithm 7.1, step 4. The estimated HSBN state-space and model parameters are returned fulfilling the objectives of the learning problem posed in Equation 7.1. Figure 7-2 illustrates the HSBN parameters produced during model initialization along with the model parameters produced by the HSBN state-space refinement algorithm in the final epoch (the results from one particular learning trial are shown here).

## 7.5 Summary

The HSBN utilizes a representation consisting of a collection of discrete states for the hidden variables. The HSBNStateSpaceEstimation algorithm was developed in this chapter to infer the set of hidden states and their associated properties. The variational Bayes lower bound is utilized as the objective to maximize in the state-space estimation algorithm. The start and

end hidden variable states are initialized to correspond to the set of observed handshape labels. The model computed using this initialization was chosen to ‘anchor’ the state-space exploration performed by the `HSBNStateSpaceEstimation` algorithm because the initial model parameters (these are also the properties of the initial latent states) are closely related to the statistics of start/end handshape sequences and their variations observed in the training set. The initial model parameters are therefore easy to interpret.

The `HSBNStateSpaceEstimation` algorithm evolves the state-space in a sequence of learning epochs. The algorithm utilizes four different state-space refinement methods denoted as `{merge-states, drop-state, reset-state, add-state}` to modify the current state-space in order to generate candidates from among which to select a suitable state-space for the next epoch. These state-space refinement methods were designed so that the changes performed by a particular state-space refinement method in a given learning epoch can be reversed using a different type of state-space refinement in a subsequent epoch thereby allowing sufficient freedom for the `HSBNStateSpaceEstimation` algorithm to explore the state-space in the local neighborhood of the initial model. Each of the refinement methods attempts to retain the association of the latent states through the learning epochs. Maintaining these associations simplifies several aspects of the learning algorithm including the forward propagation of hyper-parameters for the priors.

In the experiments chapter we analyze several aspects of the `HSBNStateSpaceEstimation` algorithm. These include the specification of appropriate priors in the initialization step, the stopping criteria (in terms of the number of learning epochs) for the learning algorithm, an assessment of the properties estimated for latent states in the final epoch and an evaluation of handshape inference performance on a sequestered test set as a function of the sequence of learning epochs.

**Algorithm 7.1:** HSBNStateSpaceEstimation: Estimate a state-space  $\hat{\mathbf{Z}}$  and associated parameters  $\lambda_{\hat{\mathbf{Z}}}$  for the HSBN

```

Inputs :  $\mathbf{x}$  Handshape label pairs for signs contained in a training set,
           :  $\mathbf{x}$  is arranged as described in Section 6.3.

Outputs:  $\hat{\mathbf{Z}}$  State-space estimated for the HSBN,
           :  $\lambda_{\hat{\mathbf{Z}}}$  HSBN parameters associated with the estimated state-space,  $\hat{\mathbf{Z}}$ .

/* Initialize the state-space and the associated hyper-parameters
using Algorithm 7.2 */
1  $(\omega_{\tau=1}, L_{\tau=1}^{\text{VB}}, \mathbf{Z}_{\tau=1}) \leftarrow \text{HSBNStateSpaceInitialization}(\mathbf{x});$ 
/* Adapt the state-space by applying a sequence of state-space
refinements */
2  $\tau \leftarrow 1;$ 
3 repeat
    /* Estimate the state-space and the parameters for the next epoch
    using Algorithm 7.3 */
4  $(\omega_{\tau+1}, L_{\tau+1}^{\text{VB}}, \mathbf{Z}_{\tau+1}) \leftarrow$ 
     $\text{HSBNStateSpaceSelection}(\mathbf{x}, \omega_{\tau}, L_{\tau}^{\text{VB}}, \mathbf{Z}_{\tau}, \omega_{\tau=1}, \mathbf{Z}_{\tau=1});$ 
5  $\tau \leftarrow \tau + 1;$ 
until the lower bound,  $L_{\tau}^{\text{VB}}$ , converges;
6  $\hat{\mathbf{Z}} \leftarrow \mathbf{Z}_{\tau};$ 
7  $\lambda_{\hat{\mathbf{Z}}} \leftarrow \mathbb{E}_{\omega_{\tau}}[\lambda];$ 

```



**Algorithm 7.2:** HSBNStateSpaceInitialization: Compute HSBN hyper-parameters in the first learning epoch

**Inputs** :  $\mathbf{x}$  Training set (as in Algorithm 7.1).  
**Outputs**:  $\omega_{\tau=1}$  HSBN hyper-parameters estimated in the first learning epoch,  
:  $L_{\tau=1}^{\text{VB}}$  VB lower bound estimated in the first learning epoch,  
:  $\mathbf{Z}_{\tau=1}$  State-space chosen in the first learning epoch.

- 1  $\mathcal{X} \leftarrow$  The set of handshake labels in  $\mathbf{x}$ ;  
/\* Compute initial estimates for the hidden variable variational distributions  $\{Q_{\mathbf{z},i}^{\circ}[z^s, z^e]\}$  \*/
- 2  $\mathcal{Z}_{\tau=1}^s \leftarrow \mathcal{X}$ ;  $\mathcal{Z}_{\tau=1}^e \leftarrow \mathcal{X}$ ;  $\mathbf{Z}_{\tau=1} \leftarrow (\mathcal{Z}_{\tau=1}^s, \mathcal{Z}_{\tau=1}^e)$ ;
- 3 **for**  $i \leftarrow 1$  to  $|\mathcal{V}_{\mathbf{x}}|$  **do**
- 4      $Q_{\mathbf{z},i}^{\circ}[z^s, z^e] \leftarrow 0$ ,  $\forall z^s \in \mathcal{Z}_{\tau=1}^s, z^e \in \mathcal{Z}_{\tau=1}^e$ ;
- 5     **for**  $j \leftarrow 1$  to  $|\mathbf{x}_i|$  **do**
- 6          $Q_{\mathbf{z},i}^{\circ}[x_{ij}^{s;\text{D}}, x_{ij}^{e;\text{D}}] \leftarrow Q_{\mathbf{z},i}^{\circ}[x_{ij}^{s;\text{D}}, x_{ij}^{e;\text{D}}] + 1$ ;
- 7          $Q_{\mathbf{z},i}^{\circ}[x_{ij}^{s;\text{N}}, x_{ij}^{e;\text{N}}] \leftarrow Q_{\mathbf{z},i}^{\circ}[x_{ij}^{s;\text{N}}, x_{ij}^{e;\text{N}}] + 1$ ;
- 8     **end**  
 $Q_{\mathbf{z},i}^{\circ}[z^s, z^e] \leftarrow \frac{Q_{\mathbf{z},i}^{\circ}[z^s, z^e]}{\sum_{z^s \in \mathcal{Z}_{\tau=1}^s, z^e \in \mathcal{Z}_{\tau=1}^e} Q_{\mathbf{z},i}^{\circ}[z^s, z^e]} \quad \forall z^s \in \mathcal{Z}_{\tau=1}^s, z^e \in \mathcal{Z}_{\tau=1}^e$ ;
- 9     **end**  
/\* The computed initial variational distributions  $Q_{\mathbf{z},i}^{\circ}$  are used to initialize the VBEM algorithm, Algorithm 6.4 \*/
- 9 **repeat**
- 10     **if** first EM iteration **then**
- 11          $Q_{\mathbf{z},i}[z^s, z^e] \leftarrow Q_{\mathbf{z},i}^{\circ}[z^s, z^e] \quad \forall z^s \in \mathcal{Z}_{\tau=1}^s, z^e \in \mathcal{Z}_{\tau=1}^e$ ;
- 12     **else**
- 13          $Q_{\mathbf{z},i}[z^s, z^e] \leftarrow$  Update using VBEM E-step, Algorithm 6.4, steps 4-5;
- 13          $L_{\mathbf{Z}}^{\text{VB}} \leftarrow$  Update the VB lower bound, Algorithm 6.4, step 6;
- 14     **end**  
 $\omega^* \leftarrow$  Update using VBEM M-step, Algorithm 6.4, steps 7-10;
- 15     **until** the lower bound,  $L_{\mathbf{Z}}^{\text{VB}}$  **and** the parameters,  $\omega^*$  converge;
- 15  $\omega_{\tau=1} \leftarrow \omega^*$ ;
- 16  $L_{\tau=1}^{\text{VB}} \leftarrow L_{\mathbf{Z}}^{\text{VB}}$ ;

**Algorithm 7.3:** HSBNStateSpaceSelection: Select state-space for the next epoch

```

Inputs :  $\mathbf{x}$  Training set (as in Algorithm 7.1),
          :  $\omega_\tau$  Hyper-parameters estimated in epoch  $\tau$ ,
          :  $L_\tau^{\text{VB}}$  VB lower bound estimated in epoch  $\tau$ ,
          :  $\mathbf{Z}_\tau = (\mathbf{Z}_\tau^s, \mathbf{Z}_\tau^e)$  Where,  $\mathbf{Z}_\tau^s \subseteq \mathbf{Z}_{\tau=1}^s$  and  $\mathbf{Z}_\tau^e \subseteq \mathbf{Z}_{\tau=1}^e$ , cur. state-space,
          :  $\omega_{\tau=1}$  Hyper-parameters estimated in the first epoch,
          :  $\mathbf{Z}_{\tau=1}$  State-space estimated in the first epoch.

Outputs:  $\mathbf{Z}_{\tau+1}$  State-space selected for the next epoch,
          :  $\omega_{\tau+1}$  Hyper-parameters after the state-space refinement,
          :  $L_{\tau+1}^{\text{VB}}$  VB lower bound after the state-space refinement.

/* Generate state-space candidates */
1 for  $\rho \in \{s, e\}$ ,  $(\psi, \varphi) \in \{\text{unique pairs of states taken from } \mathbf{Z}_\tau^\rho, \psi < \varphi\}$  do
  /* Apply MergeHSBNstates state-space refinement, Algorithm 7.4 */
2   $(\omega_{\tau+1, k}, L_{\tau+1, k}^{\text{VB}}, \mathbf{Z}_{\tau+1, k}) \leftarrow \text{MergeHSBNstates}(\mathbf{x}, \omega_\tau, \mathbf{Z}_\tau, \rho, \psi, \varphi); k++;$ 
3  for  $\rho \in \{s, e\}$ ,  $\psi \in \mathbf{Z}_\tau^\rho$  do
  /* Apply DropHSBNstate state-space refinement, Algorithm 7.5 */
4   $(\omega_{\tau+1, k}, L_{\tau+1, k}^{\text{VB}}, \mathbf{Z}_{\tau+1, k}) \leftarrow \text{DropHSBNstate}(\mathbf{x}, \omega_\tau, \mathbf{Z}_\tau, \rho, \psi); k++;$ 
5  for  $\rho \in \{s, e\}$ ,  $\psi \in \mathbf{Z}_\tau^\rho$  do
  /* Apply ResetHSBNstate state-space refinement, Algorithm 7.6 */
6   $(\omega_{\tau+1, k}, L_{\tau+1, k}^{\text{VB}}, \mathbf{Z}_{\tau+1, k}) \leftarrow \text{ResetHSBNstate}(\mathbf{x}, \omega_\tau, \mathbf{Z}_\tau, \omega_{\tau=1}, \rho, \psi); k++;$ 
7  for  $\rho \in \{s, e\}$ ,  $\psi \in \mathbf{Z}_{\tau=1}^\rho \setminus \mathbf{Z}_\tau^\rho$  do
  /* Apply AddHSBNstate state-space refinement, Algorithm 7.7 */
8   $(\omega_{\tau+1, k}, L_{\tau+1, k}^{\text{VB}}, \mathbf{Z}_{\tau+1, k}) \leftarrow \text{AddHSBNstate}(\mathbf{x}, \omega_\tau, \mathbf{Z}_\tau, \omega_{\tau=1}, \rho, \psi); k++;$ 
  /* Compute acceptance log-ratios for the state-space candidates */
9   $\tilde{\mathbf{r}}[k] \leftarrow L_{\tau+1, k}^{\text{VB}} - L_\tau^{\text{VB}}; \quad \forall k: 1 \leq k \leq |\{\omega_{\tau+1, \cdot}\}|$ 
10  $\mathbf{r} \leftarrow$  Select equal number of top candidates for different refinement types from  $\tilde{\mathbf{r}}$ ;
  /* Transform  $\mathbf{r}$  to construct an acceptance ratio distribution */
11  $\mathbf{r}[k] \leftarrow \exp\left(\frac{\mathbf{r}[k]}{\max(|\max(\mathbf{r})|, 1)}\right); \quad \mathbf{r}[k] \leftarrow \frac{\mathbf{r}[k]}{\sum_k \mathbf{r}[k]}; \quad \forall k: 1 \leq k \leq \text{len}(\mathbf{r})$ 
  /*Sample state-space candidate using acceptance ratio distribution*/
12  $l \sim \mathbf{r};$ 
13  $(\omega_{\tau+1}, L_{\tau+1}^{\text{VB}}, \mathbf{Z}_{\tau+1}) \leftarrow (\omega_{\tau+1, l}, L_{\tau+1, l}^{\text{VB}}, \mathbf{Z}_{\tau+1, l});$ 

```

**Algorithm 7.4:** MergeHSBNstates: Merge selected pair of HSBN hidden variable states

**Inputs** :  $\mathbf{x}$  Training set (as in Algorithm 7.1),  
:  $\omega_\tau$  Hyper-parameters estimated in epoch  $\tau$ ,  
:  $\mathbf{Z}_\tau = (\mathbf{Z}_\tau^s, \mathbf{Z}_\tau^e)$  Where,  $\mathbf{Z}_\tau^s \subseteq \mathbf{Z}_{\tau=1}^s$  and  $\mathbf{Z}_\tau^e \subseteq \mathbf{Z}_{\tau=1}^e$ , cur. state-space,  
:  $\rho \in \{s, e\}$  Select start or end hidden variable states,  
:  $\psi, \varphi \in \mathbf{Z}_\tau^\rho$  Selected indices of start/end hidden states to merge.

**Outputs:**  $\omega_\tau^{\text{merge}: \rho, \psi \leftarrow \varphi}$  Hyper-parameters after the state-space refinement,  
:  $L_\tau^{\text{merge}: \rho, \psi \leftarrow \varphi}$  VBEM lower bound after the state-space refinement,  
:  $\mathbf{Z}_\tau^{\text{merge}: \rho, \psi \leftarrow \varphi}$  State-space after applying the state-space refinement.

/\* Merge specified  $\psi, \varphi$  rows/columns in the hyper-parameter arrays \*/

1 **if**  $\rho == s$  **then**

2      $(\tilde{\mathbf{Z}}^s, \tilde{\mathbf{Z}}^e) \leftarrow (\mathbf{Z}_\tau^s \setminus \{\varphi\}, \mathbf{Z}_\tau^e);$

3      $\tilde{\nu}[z^s] \leftarrow \nu_\tau[z^s] \quad \forall z^s \in \tilde{\mathbf{Z}}^s;$

4      $\tilde{\alpha}[z^s, z^e] \leftarrow \alpha_\tau[z^s, z^e] \quad \forall z^s \in \tilde{\mathbf{Z}}^s, z^e \in \tilde{\mathbf{Z}}^e;$

5      $\tilde{\beta}^s[z^s, x] \leftarrow \beta_\tau^s[z^s, x] \quad \forall z^s \in \tilde{\mathbf{Z}}^s, x \in \mathcal{X};$

6      $\tilde{\beta}^e[z^e, x] \leftarrow \beta_\tau^e[z^e, x] \quad \forall z^e \in \tilde{\mathbf{Z}}^e, x \in \mathcal{X};$

7      $\tilde{\nu}[\psi] \leftarrow \nu_\tau[\psi] + \nu_\tau[\varphi];$

8      $\tilde{\alpha}[\psi, z^e] \leftarrow \alpha_\tau[\psi, z^e] + \alpha_\tau[\varphi, z^e] \quad \forall z^e \in \tilde{\mathbf{Z}}^e;$

9      $\tilde{\beta}^s[\psi, x] \leftarrow \beta_\tau^s[\psi, x] + \beta_\tau^s[\varphi, x] \quad \forall x \in \mathcal{X};$

**else**

10      $(\tilde{\mathbf{Z}}^s, \tilde{\mathbf{Z}}^e) \leftarrow (\mathbf{Z}_\tau^s, \mathbf{Z}_\tau^e \setminus \{\varphi\});$

11      $\tilde{\nu}[z^s] \leftarrow \nu_\tau[z^s] \quad \forall z^s \in \tilde{\mathbf{Z}}^s;$

12      $\tilde{\alpha}[z^s, z^e] \leftarrow \alpha_\tau[z^s, z^e] \quad \forall z^s \in \tilde{\mathbf{Z}}^s, z^e \in \tilde{\mathbf{Z}}^e;$

13      $\tilde{\beta}^s[z^s, x] \leftarrow \beta_\tau^s[z^s, x] \quad \forall z^s \in \tilde{\mathbf{Z}}^s, x \in \mathcal{X};$

14      $\tilde{\beta}^e[z^e, x] \leftarrow \beta_\tau^e[z^e, x] \quad \forall z^e \in \tilde{\mathbf{Z}}^e, x \in \mathcal{X};$

15      $\tilde{\alpha}[z^s, \psi] \leftarrow \alpha_\tau[z^s, \psi] + \alpha_\tau[z^s, \varphi] \quad \forall z^s \in \tilde{\mathbf{Z}}^s;$

16      $\tilde{\beta}^e[\psi, x] \leftarrow \beta_\tau^e[\psi, x] + \beta_\tau^e[\varphi, x] \quad \forall x \in \mathcal{X};$

**end**

/\* The transformed hyper-parameter candidates,  $\tilde{\omega}$ , are used to initialize the VBEM algorithm, Algorithm 6.4 \*/

17      $(\omega_\tau^{\text{merge}: \rho, \psi \leftarrow \varphi}, L_\tau^{\text{merge}: \rho, \psi \leftarrow \varphi}) \leftarrow \text{VBEM}^{\text{congruent}}(\mathbf{x}, \omega^\circ = \{\tilde{\nu}, \tilde{\alpha}, \tilde{\beta}^s, \tilde{\beta}^e\});$

18      $\mathbf{Z}_\tau^{\text{merge}: \rho, \psi \leftarrow \varphi} \leftarrow (\tilde{\mathbf{Z}}^s, \tilde{\mathbf{Z}}^e);$

**Algorithm 7.5:** DropHSBNstate: Removes a state from the HSBN hidden variable state-space

<b>Inputs</b>	: $\mathbf{x}$	Training set (as in Algorithm 7.1),
	: $\omega_\tau$	Hyper-parameters estimated in epoch $\tau$ ,
	: $\mathbf{Z}_\tau = (\mathbf{Z}_\tau^s, \mathbf{Z}_\tau^e)$	Where, $\mathbf{Z}_\tau^s \subseteq \mathbf{Z}_{\tau=1}^s$ and $\mathbf{Z}_\tau^e \subseteq \mathbf{Z}_{\tau=1}^e$ , cur. state-space,
	: $\rho \in \{s, e\}$	Start or end hidden variable state selected to drop,
	: $\psi \in \mathcal{Z}_\tau^\rho$	Selected index of start/end hidden state to drop.

<b>Outputs:</b>	$\omega_\tau^{\text{drop}:\rho,\psi}$	Hyper-parameters after the state-space refinement,
	$L_\tau^{\text{drop}:\rho,\psi}$	VBEM lower bound after the state-space refinement,
	$\mathbf{Z}_\tau^{\text{drop}:\rho,\psi}$	State-space after applying the state-space refinement.

```

/* Remove  $\psi^{\text{th}}$  row or  $\psi^{\text{th}}$  column from the hyper-parameter arrays */
1  if  $\rho == s$  then
2       $(\tilde{\mathbf{Z}}^s, \tilde{\mathbf{Z}}^e) \leftarrow (\mathbf{Z}_\tau^s \setminus \{\psi\}, \mathbf{Z}_\tau^e)$ ;
3       $\tilde{\nu}[z^s] \leftarrow \nu_\tau[z^s] \quad \forall z^s \in \tilde{\mathbf{Z}}^s$ ;
4       $\tilde{\alpha}[z^s, z^e] \leftarrow \alpha_\tau[z^s, z^e] \quad \forall z^s \in \tilde{\mathbf{Z}}^s, z^e \in \tilde{\mathbf{Z}}^e$ ;
5       $\tilde{\beta}^s[z^s, x] \leftarrow \beta_\tau^s[z^s, x] \quad \forall z^s \in \tilde{\mathbf{Z}}^s, x \in \mathcal{X}$ ;
6       $\tilde{\beta}^e[z^e, x] \leftarrow \beta_\tau^e[z^e, x] \quad \forall z^e \in \tilde{\mathbf{Z}}^e, x \in \mathcal{X}$ ;
    else
7       $(\tilde{\mathbf{Z}}^s, \tilde{\mathbf{Z}}^e) \leftarrow (\mathbf{Z}_\tau^s, \mathbf{Z}_\tau^e \setminus \{\psi\})$ ;
8       $\tilde{\nu}[z^s] \leftarrow \nu_\tau[z^s] \quad \forall z^s \in \tilde{\mathbf{Z}}^s$ ;
9       $\tilde{\alpha}[z^s, z^e] \leftarrow \alpha_\tau[z^s, z^e] \quad \forall z^s \in \tilde{\mathbf{Z}}^s, z^e \in \tilde{\mathbf{Z}}^e$ ;
10      $\tilde{\beta}^s[z^s, x] \leftarrow \beta_\tau^s[z^s, x] \quad \forall z^s \in \tilde{\mathbf{Z}}^s, x \in \mathcal{X}$ ;
11      $\tilde{\beta}^e[z^e, x] \leftarrow \beta_\tau^e[z^e, x] \quad \forall z^e \in \tilde{\mathbf{Z}}^e, x \in \mathcal{X}$ ;
    end

/* The transformed hyper-parameter candidates,  $\tilde{\omega}$ , are used to
initialize the VBEM algorithm, Algorithm 6.4 */
12   $(\omega_\tau^{\text{drop}:\rho,\psi}, L_\tau^{\text{drop}:\rho,\psi}) \leftarrow \text{VBEM}^{\text{congruent}}(\mathbf{x}, \omega^\circ = \{\tilde{\nu}, \tilde{\alpha}, \tilde{\beta}^s, \tilde{\beta}^e\})$ ;
13   $\mathbf{Z}_\tau^{\text{drop}:\rho,\psi} \leftarrow (\tilde{\mathbf{Z}}^s, \tilde{\mathbf{Z}}^e)$ ;

```

**Algorithm 7.6:** ResetHSBNstate: Resets the values of hyper-parameters for a selected HSBN hidden variable state to corresponding hyper-parameter values from the first epoch,  $\omega_{\tau=1}$

**Inputs** :  $\mathbf{x}$  Training set (as in Algorithm 7.1),  
:  $\omega_{\tau}$  Hyper-parameters estimated in epoch  $\tau$ ,  
:  $\mathbf{Z}_{\tau} = (\mathcal{Z}_{\tau}^s, \mathcal{Z}_{\tau}^e)$  Where,  $\mathcal{Z}_{\tau}^s \subseteq \mathcal{Z}_{\tau=1}^s$  and  $\mathcal{Z}_{\tau}^e \subseteq \mathcal{Z}_{\tau=1}^e$ , cur. state-space  
:  $\omega_{\tau=1}$  Hyper-parameters estimated in the first epoch,  
:  $\rho \in \{s, e\}$  Start or end hidden variable state selected to reset,  
:  $\psi \in \mathcal{Z}_{\tau}^{\rho}$  Selected index of start/end hidden state to reset.

**Outputs**:  $\omega_{\tau}^{\text{reset}:\rho,\psi}$  Hyper-parameters after the state-space refinement,  
:  $L_{\tau}^{\text{reset}:\rho,\psi}$  VBEM lower bound after the state-space refinement,  
:  $\mathbf{Z}_{\tau}^{\text{reset}:\rho,\psi}$  State-space after applying the state-space refinement.

- 1  $\tilde{\nu}[z^s] \leftarrow \nu_{\tau}[z^s] \quad \forall z^s \in \mathcal{Z}_{\tau}^s;$
- 2  $\tilde{\alpha}[z^s, z^e] \leftarrow \alpha_{\tau}[z^s, z^e] \quad \forall z^s \in \mathcal{Z}_{\tau}^s, z^e \in \mathcal{Z}_{\tau}^e;$
- 3  $\tilde{\beta}^s[z^s, x] \leftarrow \beta_{\tau}^s[z^s, x] \quad \forall z^s \in \mathcal{Z}_{\tau}^s, x \in \mathcal{X};$
- 4  $\tilde{\beta}^e[z^e, x] \leftarrow \beta_{\tau}^e[z^e, x] \quad \forall z^e \in \mathcal{Z}_{\tau}^e, x \in \mathcal{X};$

/\* Replace the values of  $\psi^{\text{th}}$  row or  $\psi^{\text{th}}$  column in  $\tilde{\omega}$  with values from the initial hyper-parameter arrays ( $\omega_{\tau=1}$ ) \*/

- 5 **if**  $\rho == s$  **then**
- 6      $\tilde{\nu}[\psi] \leftarrow \nu_{\tau=1}[\psi];$
- 7      $\tilde{\alpha}[\psi, z^e] \leftarrow \alpha_{\tau=1}[\psi, z^e] \quad \forall z^e \in \mathcal{Z}_{\tau}^e;$
- 8      $\tilde{\beta}^s[\psi, x] \leftarrow \beta_{\tau=1}^s[\psi, x] \quad \forall x \in \mathcal{X};$
- else**
- 9      $\tilde{\alpha}[z^s, \psi] \leftarrow \alpha_{\tau=1}[z^s, \psi] \quad \forall z^s \in \mathcal{Z}_{\tau}^s;$
- 10      $\tilde{\beta}^e[\psi, x] \leftarrow \beta_{\tau=1}^e[\psi, x] \quad \forall x \in \mathcal{X};$
- end**

/\* The transformed hyper-parameter candidates,  $\tilde{\omega}$ , are used to initialize the VBEM algorithm, Algorithm 6.4 \*/

- 11  $(\omega_{\tau}^{\text{reset}:\rho,\psi}, L_{\tau}^{\text{reset}:\rho,\psi}) \leftarrow \text{VBEM}^{\text{congruent}}(\mathbf{x}, \omega^{\circ} = \{\tilde{\nu}, \tilde{\alpha}, \tilde{\beta}^s, \tilde{\beta}^e\});$
- 12  $\mathbf{Z}_{\tau}^{\text{reset}:\rho,\psi} \leftarrow (\mathcal{Z}_{\tau}^s, \mathcal{Z}_{\tau}^e);$

**Algorithm 7.7:** AddHSBNstate: Adds a state to the HSBN hidden variable state-space by splicing in values for hyper-parameters from the first epoch,  $\omega_{\tau=1}$

**Inputs** :  $\mathbf{x}$  Training set (as in Algorithm 7.1),  
:  $\omega_{\tau}$  Hyper-parameters estimated in epoch  $\tau$ ,  
:  $\mathbf{Z}_{\tau} = (\mathcal{Z}_{\tau}^s, \mathcal{Z}_{\tau}^e)$  Where,  $\mathcal{Z}_{\tau}^s \subseteq \mathcal{Z}_{\tau=1}^s$  and  $\mathcal{Z}_{\tau}^e \subseteq \mathcal{Z}_{\tau=1}^e$ , cur. state-space  
:  $\omega_{\tau=1}$  Hyper-parameters estimated in the first epoch,  
:  $\rho \in \{s, e\}$  Start or end hidden variable state selected to add,  
:  $\psi \in \mathcal{Z}_{\tau=1}^{\rho} \setminus \mathcal{Z}_{\tau}^{\rho}$  Selected index of start/end hidden state to add.

**Outputs**:  $\omega_{\tau}^{\text{add}:\rho,\psi}$  Hyper-parameters after the state-space refinement,  
:  $L_{\tau}^{\text{add}:\rho,\psi}$  VBEM lower bound after the state-space refinement,  
:  $\mathbf{Z}_{\tau}^{\text{add}:\rho,\psi}$  State-space after applying the state-space refinement.

- 1  $\tilde{\nu}[z^s] \leftarrow \nu_{\tau}[z^s] \quad \forall z^s \in \mathcal{Z}_{\tau}^s;$
- 2  $\tilde{\alpha}[z^s, z^e] \leftarrow \alpha_{\tau}[z^s, z^e] \quad \forall z^s \in \mathcal{Z}_{\tau}^s, z^e \in \mathcal{Z}_{\tau}^e;$
- 3  $\tilde{\beta}^s[z^s, x] \leftarrow \beta_{\tau}^s[z^s, x] \quad \forall z^s \in \mathcal{Z}_{\tau}^s, x \in \mathcal{X};$
- 4  $\tilde{\beta}^e[z^e, x] \leftarrow \beta_{\tau}^e[z^e, x] \quad \forall z^e \in \mathcal{Z}_{\tau}^e, x \in \mathcal{X};$

/\* Splice the  $\psi^{\text{th}}$  row or  $\psi^{\text{th}}$  column from the initial hyper-parameter arrays ( $\omega_{\tau=1}$ ) into the transformed hyper-parameter arrays ( $\tilde{\omega}$ ) \*/

- 5 **if**  $\rho == s$  **then**
- 6      $(\tilde{\mathcal{Z}}^s, \tilde{\mathcal{Z}}^e) \leftarrow (\mathcal{Z}_{\tau}^s \cup \{\psi\}, \mathcal{Z}_{\tau}^e);$
- 7      $\tilde{\nu}[\psi] \leftarrow \nu_{\tau=1}[\psi];$
- 8      $\tilde{\alpha}[\psi, z^e] \leftarrow \alpha_{\tau=1}[\psi, z^e] \quad \forall z^e \in \tilde{\mathcal{Z}}^e;$
- 9      $\tilde{\beta}^s[\psi, x] \leftarrow \beta_{\tau=1}^s[\psi, x] \quad \forall x \in \mathcal{X};$
- else
- 10      $(\tilde{\mathcal{Z}}^s, \tilde{\mathcal{Z}}^e) \leftarrow (\mathcal{Z}_{\tau}^s, \mathcal{Z}_{\tau}^e \cup \{\psi\});$
- 11      $\tilde{\alpha}[z^s, \psi] \leftarrow \alpha_{\tau=1}[z^s, \psi] \quad \forall z^s \in \tilde{\mathcal{Z}}^s;$
- 12      $\tilde{\beta}^e[\psi, x] \leftarrow \beta_{\tau=1}^e[\psi, x] \quad \forall x \in \mathcal{X};$
- end

/\* The transformed hyper-parameter candidates,  $\tilde{\omega}$ , are used to initialize the VBEM algorithm, Algorithm 6.4 \*/

- 13  $(\omega_{\tau}^{\text{add}:\rho,\psi}, L_{\tau}^{\text{add}:\rho,\psi}) \leftarrow \text{VBEM}^{\text{congruent}}(\mathbf{x}, \omega^{\circ} = \{\tilde{\nu}, \tilde{\alpha}, \tilde{\beta}^s, \tilde{\beta}^e\});$
- 14  $\mathbf{Z}_{\tau}^{\text{add}:\rho,\psi} \leftarrow (\tilde{\mathcal{Z}}^s, \tilde{\mathcal{Z}}^e);$

## Chapter 8

# Handshape image observation likelihood model

The observation likelihoods in the HSBN based handshape inference formulation (developed in Chapter 5) are represented in the posterior form as,  $P(X = x | I = \mathbf{i})$ . Given image,  $\mathbf{i}$ , of a handshape in an input video at either the start or end points of a sign, we need the likelihoods of different handshape classes,  $x \in \mathcal{X}$ .

Variations in the image appearance of handshapes makes estimation of the observation likelihood challenging. Some sources of variation in handshape appearance include differences in the 3D orientation of the hands, differences in the anthropometric properties, differences in how a handshape is articulated either by the same signer or by different signers, and differences in handshape production influenced by the phonological environment within which the handshape appears in a sign. Image clutter is another issue that makes the estimation of the observation likelihood for hand images in sign language video a challenge. Even after skin color based segmentation these images include considerable amounts of clutter because the hands are frequently articulated close to the face or to the other hand.

We adopt a data-driven approach in this work wherein a collection of annotated start/end handshapes obtained from several native signers serves as a database for the purposes of retrieving handshape matches. We start by describing the proposed model for constructing an observation likelihood given the retrieved handshape images and their annotated handshape labels. We then develop an algorithm for non-rigid image alignment to incorporate robustness to some of the variations described above during handshape retrieval.

### 8.1 Computing the handshape observation likelihood

A database of hand images  $\{\mathbf{i}_{\text{DB}}^i\}$  annotated with handshape labels  $\{x_{\text{DB}}^i\}$  is obtained by collecting start/end handshapes for signs contained in the training set. A method to compute the similarity score,  $\text{sim}(\mathbf{i}, \mathbf{j})$ , for handshape image pairs is assumed in computing the observation likelihoods. Given an image  $\mathbf{i}$  in the query sign, its  $K$ -nearest neighbors from the database ranked in decreasing order of appearance similarity are denoted as,  $\{\mathbf{i}_{\text{DB}}^k, x_{\text{DB}}^k\}$ ,  $1 \leq k \leq K$ . The retrieved handshape labels are then used in computing the observation likelihoods using the following expression,

$$P(X = x | I = \mathbf{i}) = \frac{1}{C_P} \sum_{k=1}^K e^{-\beta(k-1)} \delta(x_{\text{DB}}^k, x). \quad (8.1)$$

Here,  $\beta, K$  are predefined parameters for the handshape inference algorithm,  $C_P$  is a normalizing constant and  $\delta$  is the indicator function. The handshape match,  $x_{\text{DB}}^k$ , retrieved at rank  $k$  contributes a score  $e^{-\beta k}$  to the observation likelihood for the label  $x_{\text{DB}}^k$ .

Unlike in a conventional  $k$ -NN density estimator wherein the similarity score, or a distance measure,  $\text{sim}(\mathbf{i}, \mathbf{j})$  appears in the exponent, the above expression for the handshape observation likelihood employs the retrieved ranks for the handshape labels. This is because certain regularity properties satisfied by the similarity scores employed in conventional  $k$ -NN methods (such as the properties required for the underlying distance to define a metric) are violated by the alignment based methods employed here in computing a similarity score for handshape image pairs. Utilizing the retrieved rank for handshape labels was therefore observed to yield more predictable results in our empirical evaluation. The rank based formulation also simplifies the comparison of handshape inference accuracies obtained using different types of image alignment methods.

### 8.2 Computing the handshape appearance similarity score, $\text{sim}(\mathbf{i}, \mathbf{j})$

Given an input image of a handshape,  $\mathbf{i}$ , we use a similarity scoring function  $\text{sim}(\mathbf{i}, \mathbf{j})$  to retrieve similar handshapes from our annotated database. To provide robustness to variations in the handshape image appearances, a non-rigid image alignment is needed.



### 8.2.1 Background

To align an image pair,  $(\mathbf{i}, \mathbf{j})$ , we compute the vectors  $\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}$  that map feature locations in image  $\mathbf{i}$  to pixel locations in image  $\mathbf{j}$  by minimizing an image alignment cost comprising two terms: (a) the data association cost that measures the accuracy of the predicted registration in aligning local image features, and (b) the spatial prior that imposes a smoothness constraint on the estimated alignment vectors. The alignment cost minimization can therefore be formulated in general terms as follows,

$$\begin{aligned} \mathbf{a}^{*\mathbf{i} \rightarrow \mathbf{j}} &= \arg \min_{\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}} \left[ E_{\text{align}}(\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}) \right] \\ &= \arg \min_{\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}} \left[ E_{\text{data-association}}(\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}) + E_{\text{spatial-smoothness}}(\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}) \right]. \end{aligned} \quad (8.2)$$

The alignment vectors,  $\mathbf{a}^{\mathbf{j} \rightarrow \mathbf{i}}$ , in the converse direction are computed in a similar fashion. Specific choices for the data-association cost and smoothness prior terms are presented here towards developing a computationally efficient `HandshapelImageAlignment` algorithm for computing non-rigid alignments for handshape image pairs.

Solving for the global minimum of the total alignment cost,  $E_{\text{align}}(\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}})$ , is typically intractable as this minimization corresponds to a NP-hard MAP estimation problem in general MRFs [Felzenszwalb and Zabih, 2011], and therefore many approximate methods to minimize the cost have been proposed in the literature. These approaches fall into two broad categories: approaches based on the message passing algorithm (Loopy Belief Propagation or LBP) [Liu et al., 2008] and approaches based on solving sparse linear system of equations (LSEs) [Pilet et al., 2008, Huang et al., 2006].

Loopy Belief Propagation approaches typically assume a discrete label set for the alignment vectors. A quantization of the alignment vectors using a locally sampled set of feature locations in the spatial neighborhood of each control lattice location yields the label set containing the alignment candidates. Even though LBP is widely used with several types of spatial smoothness priors (examples include non-convex priors as are often used for optical flow computation [Liu et al., 2008]), the algorithm remains computationally expensive. The message passing cost in the case of general smoothness priors scales quadratically in the

label set size,  $|\mathcal{W}|$ . A large number of message passing iterations is also typically needed for message passing algorithms to converge to a stable solution. This computational cost precludes using a large densely sampled local search neighborhood in computing alignments for handshape image pairs.

Linear system of equations based approaches employ spatial smoothness priors from the Free Form Deformation (FFD) family. The Thin Plate Spline (TPS) [Huang et al., 2006] and the spring mesh system are two such examples. FFD priors are defined as quadratic functions of the predicted displacements:  $E_{\text{spatial-smoothness}}(\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}) = \mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}T} \mathbf{K} \mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}$ . The FFD prior is parameterized by the stiffness matrix,  $\mathbf{K}$ . The quadratic form of the smoothness term admits an efficient gradient descent solution, which involves solving a sequence of sparse linear systems of equations (LSEs). To help circumvent the problems of local minima, [Huang et al., 2006] propose a coarse-to-fine refinement of the control lattice, while, [Pilet et al., 2008] develop a RANSAC [Fischler and Bolles, 1981] based approach. Huang et al. develop their approach in the context of aligning contours in 2D and mesh based surface representations in 3D. Pilet et al.’s method requires that the inputs possess distinctive local image features to allow the representation of the input images using a sparse set of feature point based descriptors; the authors demonstrate results for the problem of flexible 2D surface detection/re-identification. In handshape images, however, a significant portion of the handshape appearance information is contained within the handshape silhouette, a boundary contour or a sparse feature representation is therefore insufficient to capture the internal details of handshape appearance. In the proposed approach, we regard each pixel in the foreground region as containing information that is potentially useful for the task of handshape matching.

### 8.2.2 Proposed formulation for non-rigid image alignment

The `HandshapeImageAlignment` algorithm computes alignment vectors that map lattice coordinates in one image to pixel coordinates in the other image. The proposed algorithm retains the computational efficiency afforded by the LSE formulation by employing a smoothness

prior from the Free Form Deformation (FFD) family. One key distinguishing aspect of the proposed algorithm is that it iteratively adapts the smoothness prior to accommodate the different amounts of displacements in different regions of the image. This is accomplished by modifying the stiffness values for individual springs that comprise the smoothness prior based on the predicted displacements computed at each node. A randomization step as in the RANSAC algorithm is utilized to improve the robustness of the algorithm to local minima.

### Representation chosen for the alignment vectors

We choose a lattice of regularly sampled control points,  $\mathcal{G}^i = \{\mathcal{G}_{k,l}^i\} = \left\{ \left[ \mathcal{G}_{k,l;X}^i, \mathcal{G}_{k,l;Y}^i \right] \right\}$ , as feature locations to extract local image descriptors for the image  $\mathbf{i}$ . The corresponding lattice of control points in image  $\mathbf{j}$  is denoted as  $\mathcal{G}^j$ . The vectors,  $\mathbf{a}^{i \rightarrow j}$ , computed by the proposed algorithm map lattice coordinates,  $\mathcal{G}_{k,l}^i$ , in image  $\mathbf{i}$  to pixel coordinates,  $\mathcal{P}_{x,y}^j$ , in image  $\mathbf{j}$ ,

$$\mathbf{a}_{k,l}^{i \rightarrow j} : \mathcal{G}_{k,l}^i \rightarrow \mathcal{P}_{x,y}^j - \mathcal{G}_{k,l}^j, \quad \forall \mathcal{G}_{k,l}^i \in \mathcal{G}^i. \quad (8.3)$$

An expanded representation for the alignment vectors is therefore written as,

$$\mathbf{a}^{i \rightarrow j} = \left[ \mathbf{a}_{k,l}^{i \rightarrow j} \right], \quad \text{with,} \quad \mathbf{a}_{k,l}^{i \rightarrow j} = \left[ \mathbf{a}_{k,l;X}^{i \rightarrow j}, \mathbf{a}_{k,l;Y}^{i \rightarrow j} \right]. \quad (8.4)$$

### Local feature descriptors

The local image descriptors at a given feature location in the image are computed using the Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005] method. To accommodate differences in the in-plane orientations of the hands, the HOG descriptors at a given feature location are computed for a sampled set of local image orientations,  $\boldsymbol{\theta} = \{\theta\}$ . The HOG descriptors in image  $\mathbf{i}$  computed at the control lattice locations,  $\mathcal{G}_{k,l}^i$ , are denoted as  $\mathbf{h}_{k,l}^{i,\theta=0}$ . The descriptors computed in image  $\mathbf{j}$  at the coordinates,  $\mathcal{G}_{k,l;m,n}^j, \mathcal{G}_{x,y}^j$ , are denoted as,  $\mathbf{h}_{k,l;m,n}^{j,\theta}, \mathbf{h}_{x,y}^{j,\theta}, \forall \theta \in \boldsymbol{\theta}$ .

The online computation of HOG features can be substantially sped up by pre-computing summed-area tables for each of the different in-plane orientation angles,  $\theta$ , and for each of

the different gradient orientation bins that are employed in computing the HOG feature representation.

### Data-association cost term

Given the local image descriptors described above, the data-association cost for a pair of feature locations  $(\mathcal{G}_{k,l}^i, \mathcal{G}_{k,l:m,n}^j)$  is computed by searching for the best feature match among the sampled set of local orientations,

$$E_{\text{data-association}}(\mathcal{G}_{k,l}^i, \mathcal{G}_{k,l:m,n}^j) = \min_{\theta \in \Theta} \left\| \mathbf{h}_{k,l}^{i,\theta=0} - \mathbf{h}_{k,l:m,n}^{j,\theta} \right\|. \quad (8.5)$$

The HandshapelImageAlignment algorithm computes a solution to the alignment problem in a sequence of iterations indexed by  $\tau$ . The total data-association cost for an alignment,  $\mathbf{a}^{\tau:i \rightarrow j}$ , is given by,

$$E_{\text{data-association}}(\mathbf{a}^{\tau:i \rightarrow j}) = \sum_{k,l \in \mathcal{G}^i} E_{\text{data-association}}(\mathcal{G}_{k,l}^i, \mathcal{G}_{k,l}^j + \mathbf{a}_{k,l}^{\tau:i \rightarrow j}). \quad (8.6)$$

### Computing local displacements by minimizing the data-association cost

Given the alignment,  $\mathbf{a}^{\tau:i \rightarrow j}$ , computed in the current iteration the local displacements for the next iteration,  $\Delta \mathbf{a}^{\tau:i \rightarrow j}$ , are computed as,

$$\Delta \mathbf{a}_{k,l}^{\tau:i \rightarrow j} = \arg \min_{\mathcal{G}_{k,l:m,n}^j \in \mathcal{W}_{k,l}^{\tau:j}} \left[ E_{\text{data-association}}(\mathcal{G}_{k,l}^i, \mathcal{G}_{k,l:m,n}^j) \right] - \mathcal{G}_{k,l}^{\tau:j}, \quad \forall \mathcal{G}_{k,l}^i \in \mathcal{G}^i. \quad (8.7)$$

Here,  $\mathcal{G}^{\tau:j}$  are the updated control lattice coordinates in image  $\mathbf{j}$  that are obtained by incorporating the current alignment:  $\mathcal{G}_{k,l}^{\tau:j} = \mathcal{G}_{k,l}^j + \mathbf{a}_{k,l}^{\tau:i \rightarrow j}$ . The feature locations for the above minimization are situated on a local neighborhood grid; these are denoted as,  $\mathcal{W}_{k,l}^{\tau:j} = \{\mathcal{G}_{k,l:m,n}^{\tau:j}\}$ .

The local displacements,  $\Delta \mathbf{a}^{\tau:i \rightarrow j}$ , are subsequently used to compute the refined alignment,  $\mathbf{a}^{\tau+1:i \rightarrow j}$ , by incorporating the spatial smoothness term:  $\Delta \mathbf{a}^{\tau:i \rightarrow j} \rightsquigarrow \widetilde{\Delta \mathbf{a}}^{\tau:i \rightarrow j} \rightsquigarrow \mathbf{a}^{\tau+1:i \rightarrow j} = \mathbf{a}^{\tau:i \rightarrow j} + \widetilde{\Delta \mathbf{a}}^{\tau:i \rightarrow j}$ .

### Spatial smoothness term

A spring mesh system connecting pairs of nodes in the lattice is chosen to yield the following quadratic spatial smoothness cost,

$$E_{\text{spatial-smoothness}}(\mathbf{a}^{\tau:i \rightarrow j}) = \mathbf{a}^{\tau:i \rightarrow j T} \mathbf{K}^{\tau:i \rightarrow j} \mathbf{a}^{\tau:i \rightarrow j}. \quad (8.8)$$

The smoothness term is parameterized by a global stiffness matrix,  $\mathbf{K}^{\tau:i \rightarrow j}$ , obtained by assembling several local stiffness matrices,  $\mathbf{k}_{\mathbf{s},\mathbf{t}}^{\tau:i \rightarrow j}$ , each of which is associated with a spring connecting a pair of nodes ( $\mathcal{G}_{\mathbf{s}}^{\tau:j}, \mathcal{G}_{\mathbf{t}}^{\tau:j}$ ) in the control lattice. These terms include the iteration index,  $\tau$ , to denote that they are updated in each iteration of the HandshapeImageAlignment algorithm. The stiffness matrices  $\mathbf{k}_{\mathbf{s},\mathbf{t}}^{\tau}$  are defined as,

$$\mathbf{k}_{\mathbf{s},\mathbf{t}}^{\tau} = \frac{\kappa_{\mathbf{s},\mathbf{t}}^{\tau}}{\text{len}(\mathbf{s}, \mathbf{t})} \begin{bmatrix} \cos^2(\beta_{\mathbf{s},\mathbf{t}}) & \cos(\beta_{\mathbf{s},\mathbf{t}}) \sin(\beta_{\mathbf{s},\mathbf{t}}) & -\cos^2(\beta_{\mathbf{s},\mathbf{t}}) & -\cos(\beta_{\mathbf{s},\mathbf{t}}) \sin(\beta_{\mathbf{s},\mathbf{t}}) \\ \cos(\beta_{\mathbf{s},\mathbf{t}}) \sin(\beta_{\mathbf{s},\mathbf{t}}) & \sin^2(\beta_{\mathbf{s},\mathbf{t}}) & -\cos(\beta_{\mathbf{s},\mathbf{t}}) \sin(\beta_{\mathbf{s},\mathbf{t}}) & -\sin^2(\beta_{\mathbf{s},\mathbf{t}}) \\ -\cos^2(\beta_{\mathbf{s},\mathbf{t}}) & -\cos(\beta_{\mathbf{s},\mathbf{t}}) \sin(\beta_{\mathbf{s},\mathbf{t}}) & \cos^2(\beta_{\mathbf{s},\mathbf{t}}) & \cos(\beta_{\mathbf{s},\mathbf{t}}) \sin(\beta_{\mathbf{s},\mathbf{t}}) \\ -\cos(\beta_{\mathbf{s},\mathbf{t}}) \sin(\beta_{\mathbf{s},\mathbf{t}}) & -\sin^2(\beta_{\mathbf{s},\mathbf{t}}) & \cos(\beta_{\mathbf{s},\mathbf{t}}) \sin(\beta_{\mathbf{s},\mathbf{t}}) & -\sin^2(\beta_{\mathbf{s},\mathbf{t}}) \end{bmatrix}. \quad (8.9)$$

Here,  $\kappa_{\mathbf{s},\mathbf{t}}^{\tau}$  is the spring stiffness parameter,  $\text{len}(\mathbf{s}, \mathbf{t})$  and  $\beta_{\mathbf{s},\mathbf{t}}$  are the length and angle with x-axis for the spring connecting a pair of control lattice nodes, ( $\mathcal{G}_{\mathbf{s}}^{\tau:j}, \mathcal{G}_{\mathbf{t}}^{\tau:j}$ ).

### Proposed algorithm for handshape image alignment

Given the data-association and spatial smoothness terms as defined above, we now formulate the proposed algorithm to solve for the image alignment by minimizing the alignment cost. For the rest of this discussion we focus on computing the forward alignment vectors,  $\mathbf{a}^{i \rightarrow j}$  (and therefore drop the superscript,  $\mathbf{i} \rightarrow \mathbf{j}$ ). We start with the gradient descent formulation, which suggests an iterative approach to minimize the following alignment cost,

$$E_{\text{align}}(\mathbf{a}) = E_{\text{data-association}}(\mathbf{a}) + \mathbf{a}^T \mathbf{K} \mathbf{a}. \quad (8.10)$$

Setting the gradient of the total alignment cost to 0 we obtain the following condition satisfied by a local minimum,  $\mathbf{a}^*$ , of the alignment cost function,

$$-\nabla_{\mathbf{a}} E_{\text{data-association}}(\mathbf{a}^*) = \mathbf{K} \mathbf{a}^*. \quad (8.11)$$

We note here that the RHS represents the smoothness constraint and the LHS is a direction vector that decreases the data-association cost at the current solution of the image alignment objective function. We introduce force vectors,  $\mathbf{f}$ , to represent the above equation in a standard linear form. The forces applied to the lattice coordinates in the spring mesh are defined as,

$$\mathbf{f}^* = -\nabla_{\mathbf{a}} E_{\text{data-association}}(\mathbf{a}^*). \quad (8.12)$$

Here,  $\mathbf{f} = [\mathbf{f}_{k,l}]$ , and,  $\mathbf{f}_{k,l} = [\mathbf{f}_{k,l;X}, \mathbf{f}_{k,l;Y}]$ . The local solution constraint in Equation 8.11 is written as,

$$\mathbf{f}^* = \mathbf{K} \mathbf{a}^*. \quad (8.13)$$

The `HandshapelImageAlignment` algorithm deviates from this formulation in that the spring stiffness values  $\kappa_{s,t}^\tau$  are updated to conform to the predicted local displacements, thereby providing a spatially *non-uniform* smoothness prior to accommodate the different amounts of deformation found in different regions of the handshape image(s).

Given the solution for the alignment vectors  $\mathbf{a}^\tau$  in iteration  $\tau$ , we now present the steps to compute the updated alignments,  $\mathbf{a}^{\tau+1}$ .

The local displacements,  $\Delta \mathbf{a}^\tau$ , are computed by minimizing the data-association cost term, as derived in Equation 8.7. To incorporate a degree of robustness to local minima of the alignment objective in our implementation, the selection of the minimum in Equation 8.7 is performed in a stochastic fashion by choosing from among the top  $U$  matches.

The force vectors,  $\mathbf{f}^\tau$ , are obtained by normalizing the local displacements,  $\Delta \mathbf{a}^\tau$ ,

$$\mathbf{f}_{k,l}^\tau = \frac{\Delta \mathbf{a}_{k,l}^\tau}{\|\Delta \mathbf{a}_{k,l}^\tau\|}. \quad (8.14)$$

The magnitudes of the local displacements,  $\|\Delta \mathbf{a}_{k,l}^\tau\|$ , are used to adapt the spring stiffness parameters,  $\kappa^\tau$ , in the spring mesh to match the different degrees of local displacements predicted for different locations in the control lattice. The stiffness parameter,  $\kappa_{s,t}^\tau$ , for a spring that connects the pair of nodes  $(\mathcal{G}_s^{\tau:j}, \mathcal{G}_t^{\tau:j})$  is specified to be inversely proportional to the average of the displacement magnitudes predicted for the two ends of the spring – thereby relaxing the stiffness term in regions where higher local displacements are predicted

in the current iteration,  $\tau$ . Furthermore, the springs get progressively stiffer through the iterations as the alignment solution converges to a local minimum. The spring stiffness parameters are given by,

$$\kappa_{s,t}^\tau = \min \left( \frac{2 \kappa_{\text{base}}}{\|\Delta \mathbf{a}_s^\tau\| + \|\Delta \mathbf{a}_t^\tau\|}, \kappa_{\text{max}} \right). \quad (8.15)$$

Where,  $\kappa_{\text{base}}$ , is the base spring stiffness parameter. The stiffness values,  $\kappa_{s,t}^\tau$ , are used in Equation 8.9 to compute the local stiffness matrices,  $\mathbf{k}_{s,t}^\tau$ . These local stiffness matrices are assembled to yield the global stiffness matrix,  $\mathbf{K}^\tau$ .

Given the force vectors,  $\mathbf{f}^\tau$  and the stiffness matrix  $\mathbf{K}^\tau$ , a refined alignment,  $\widetilde{\Delta \mathbf{a}}^\tau$ , is computed by solving the following linear system,

$$\mathbf{K}^\tau \widetilde{\Delta \mathbf{a}}^\tau = \mathbf{f}^\tau. \quad (8.16)$$

Since  $\mathbf{K}$  is sparse, we utilize the conjugate gradient algorithm in our implementation to solve this linear system.

The updated values for the alignment vectors,  $\mathbf{a}^{\tau+1}$ , are computed by using a line-search for the scaling parameter that minimizes the data-association cost,

$$\alpha^* = \arg \min_{\alpha \in [0, \alpha_{\text{max}}]} \left[ E_{\text{data-association}} \left( \alpha \widetilde{\Delta \mathbf{a}}^\tau + \mathbf{a}^\tau \right) \right], \quad (8.17)$$

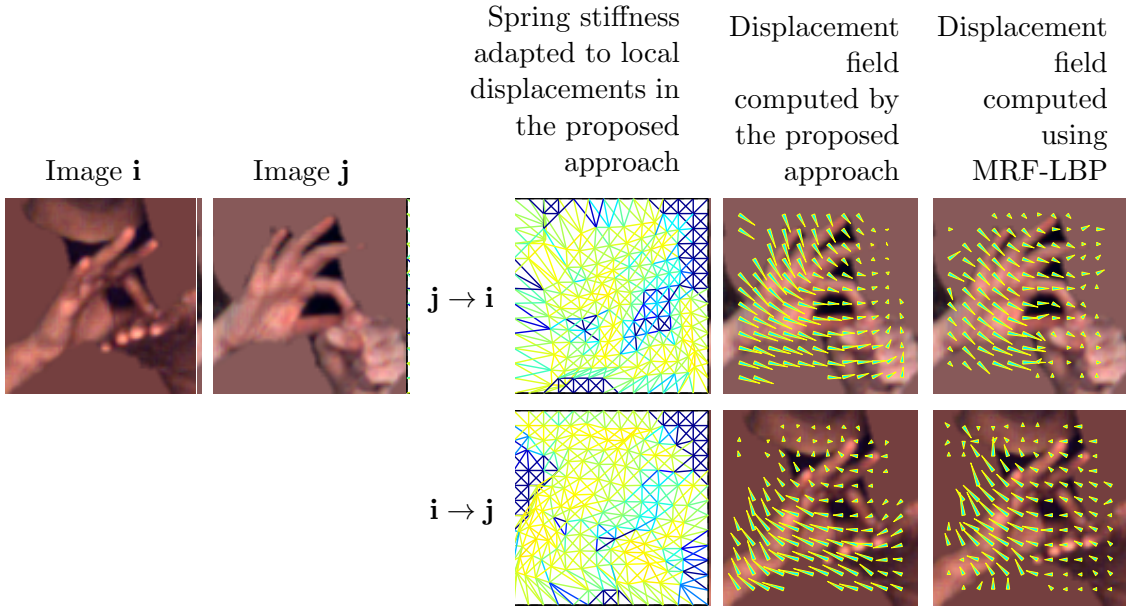
$$\mathbf{a}^{\tau+1} = \alpha^* \widetilde{\Delta \mathbf{a}}^\tau + \mathbf{a}^\tau. \quad (8.18)$$

In order to determine the initial alignments,  $\mathbf{a}^{\tau=0}$ , for the `HandshapelImageAlignment` algorithm, we solve for the affine transformation parameters utilizing the alignment vector candidates,  $\Delta \mathbf{a}^{\tau=0}$ , computed in the first iteration.

Summing the data association costs corresponding to the independently computed bi-directional alignments,  $\mathbf{a}^{*:\mathbf{i} \rightarrow \mathbf{j}}$ ,  $\mathbf{a}^{*:\mathbf{j} \rightarrow \mathbf{i}}$ , yields the appearance based similarity score for the handshape image pair,

$$\text{sim}(\mathbf{i}, \mathbf{j}) = E_{\text{data-association}}(\mathbf{a}^{*:\mathbf{i} \rightarrow \mathbf{j}}) + E_{\text{data-association}}(\mathbf{a}^{*:\mathbf{j} \rightarrow \mathbf{i}}). \quad (8.19)$$

The complete `HandshapelImageAlignment` algorithm is summarized in Algorithm 8.1.



**Figure 8-1:** Computing a bi-directional alignment for an example handshape image pair  $(i, j)$ . The displacement fields computed using the proposed algorithm are compared with those obtained using MRF-LBP. The same data association cost (obtained by comparing HOG features) and smoothness prior terms (given by a spring-mesh system) are employed in this comparison. While both approaches yield similar results, the proposed approach is an order of magnitude faster. The proposed approach adapts the spring stiffness values to provide higher rigidity in areas where less deformation is expected (darker colors indicate higher stiffness). Stiffness values displayed here correspond to the final iteration of the proposed approach.

### 8.2.3 Illustration of alignment results using the proposed algorithm

We show alignment results for an example hand image pair in Figure 8-1. The first column displays an image  $i$  of a handshape in a query sign obtained from the test signer. The second column displays the handshape image  $j$  from the database that was retrieved as one of the top-5 matches by using the proposed non-rigid image alignment algorithm. The top row displays the results of computing the alignment  $\mathbf{a}^{j \rightarrow i}$  using the proposed non-rigid alignment method and using the MRF-LBP approach. The bottom row displays the corresponding results for computing the alignment  $\mathbf{a}^{i \rightarrow j}$ . The third column in the two rows visualizes the inferred spring stiffness values in the final iteration of the proposed non-rigid



image alignment algorithm. In the top row we observe that the ring structure with two of the fingers is essentially rigid and hence higher stiffness values (darker link colors) are inferred within it and conversely, lower stiffness values are inferred in regions surrounding the extended fingers. The displacement field computed using the proposed approach is displayed in the fourth column. Results for the MRF-LBP approach minimizing the same alignment cost (but with a spatially uniform spring-mesh smoothness prior) are shown in the last column. In practice, while both approaches yield comparable alignment results, the proposed approach is an order of magnitude faster (2.4s vs. 58s) which allows a larger fraction of the database to be scanned during nearest neighbor search. Section 9.3.4 describes the details of the filter+refine method adopted for handshape retrieval. We demonstrate in our experiments that the proposed non-rigid image alignment method improves handshape retrieval accuracy when compared to an approach that does not include an image alignment step and an approach that incorporates an affine alignment between a pair of handshape images.

### 8.3 Summary

In this chapter we propose a specific implementation of an observation likelihood model to use within the HSBN formulation. The choice of a nearest neighbor approach as the underlying method in computing the handshape image observation likelihood was motivated by application domain considerations. There are a large number of handshape classes, many of which share similar configurations. Differences that arise as a consequence of gradient in handshape configuration need to be dealt with in computing the handshape observation likelihood. Furthermore, handshapes in video arise as projections of different 3D orientations of the hand. With an eye towards incorporating robustness to these factors, we propose a computationally efficient algorithm for non-rigid handshape image alignment. The nearest neighbor results computed using this method are used to produce observation likelihood scores that are compatible with the HSBN based handshape inference formulation.

**Algorithm 8.1:** HandshapelmageAlignment: Estimate the appearance based similarity score for handshape image pair.

```

Inputs  :  $\mathbf{i}, \mathbf{j}$       A pair of handshape images, centered, cropped, normalized
              :              to the same size with foreground segmentation.
Outputs:  $\text{sim}(\mathbf{i}, \mathbf{j})$  The similarity score for the input handshape image pair.

/* In subsequent steps we compute  $\mathbf{a}^{i \rightarrow j}$  (we drop the superscript) */
/* Initialize the control point lattice for  $\mathbf{i}, \mathbf{j}$  */
1   $\mathcal{G}^i \leftarrow \{\mathcal{G}_{k,l}^i\}; \mathcal{G}^j \leftarrow \{\mathcal{G}_{k,l}^j\};$ 
/* Initialize spring mesh connecting control point pairs  $(s, t)$  */
2   $\mathcal{S} \leftarrow \{(s, t)\};$  where,  $s, t \in \mathcal{G}^i$ 
/* Pre-compute sum-area tables for HOG features in  $\mathbf{i}, \mathbf{j}$ , Section 8.2.2 */
/* Compute the initial alignments  $\mathbf{a}^{\tau=0}$  using affine alignment */
3   $\Delta \mathbf{a}^{\tau=0} \leftarrow$  Computed as in Equation 8.7;
4   $\mathbf{a}^{\tau=0} \leftarrow$  Affine transformation computed using  $\Delta \mathbf{a}^{\tau=0}$ ;
5   $\tau \leftarrow 0$ ;
6  repeat
    /* Update control lattice for  $\mathbf{j}$  to include current alignment */
7   $\mathcal{G}_{k,l}^{\tau:j} \leftarrow \mathcal{G}_{k,l}^j + \mathbf{a}_{k,l}^\tau; \quad \forall \mathcal{G}_{k,l}^j \in \mathcal{G}^j$ 
    /* Compute candidate alignments using data-association cost */
8   $\Delta \mathbf{a}_{k,l}^\tau \leftarrow$  Computed as in Equation 8.7;
    /* Compute force vectors */
9   $\mathbf{f}_{k,l}^\tau \leftarrow \frac{\Delta \mathbf{a}_{k,l}^\tau}{\|\Delta \mathbf{a}_{k,l}^\tau\|}; \quad \forall \mathcal{G}_{k,l}^j \in \mathcal{G}^j$ 
    /* Compute spring stiffness values for the spring mesh */
10  $\kappa_{s,t}^\tau \leftarrow \min\left(\frac{2 \kappa_{\text{base}}}{\|\Delta \mathbf{a}_s^\tau\| + \|\Delta \mathbf{a}_t^\tau\|}, \kappa_{\text{max}}\right); \quad \forall (s, t) \in \mathcal{S}$ 
    /* Compute local and global stiffness matrices */
11  $\mathbf{k}_{s,t}^\tau \leftarrow$  Equation 8.9;  $\forall (s, t) \in \mathcal{S}$ 
12  $\mathbf{K}^\tau \leftarrow$  Assemble the local spring stiffness matrices  $\mathbf{k}_{s,t}^\tau$ ;
    /* Compute the updated alignment  $\mathbf{a}^{\tau+1}$  */
13  $\widetilde{\Delta \mathbf{a}}^\tau \leftarrow$  Solve:  $\mathbf{K}^\tau \widetilde{\Delta \mathbf{a}}^\tau = \mathbf{f}^\tau$ ;
14  $\alpha^* \leftarrow$  Computed as in Equation 8.17;
15  $\mathbf{a}^{\tau+1} \leftarrow \alpha^* \widetilde{\Delta \mathbf{a}}^\tau + \mathbf{a}^\tau$ ;
16  $\tau \leftarrow \tau + 1$ ;
until Until non-rigid alignments,  $\mathbf{a}^\tau$ , converge or #iterations are exceeded
/* Compute similarity score using the bi-directional alignments */
17  $\text{sim}(\mathbf{i}, \mathbf{j}) \leftarrow E_{\text{data-association}}(\mathbf{a}^{\tau^*: \mathbf{i} \rightarrow \mathbf{j}}) + E_{\text{data-association}}(\mathbf{a}^{\tau^*: \mathbf{j} \rightarrow \mathbf{i}});$ 

```

## Chapter 9

# Experiments: Implementation

Experiments were conducted to assess the usefulness of the HSBN formulation. Ranked handshape retrieval was chosen as the criterion for performance evaluation. The performance assessment was conducted for person-independent handshape recognition. The training and test sets were constructed from monomorphemic lexical signs contained in the ASLLVD; Chapter 4 summarizes the key aspects of the complete dataset as it pertains to the HSBN formulation. The learning of the HSBN is performed using the variational Bayes approach and involves estimating the hidden variable state-space along with the HSBN model parameters as described in Chapter 7. Handshape inference for query signs involves retrieving handshape matches from a database of start/end handshape images annotated with handshape labels. The retrieved handshape labels are re-ranked utilizing the HSBN model. Comparing the retrieved ranks of ground-truth labels before and after HSBN based inference enables us to quantify the improvement in recognition performance afforded by the proposed formulation.

In the following sections we describe the construction of the training and test sets, the procedure adopted for training the HSBN model, the procedure adopted to compute the observation likelihood for start/end hand images in a query sign and the procedure employed for evaluating the handshape inference performance.

### 9.1 Training, retrieval and test sets for HSBN evaluation

Among the several sign language datasets available for SLR research, the ASLLVD is unique in that it provides a reasonably large collection of signs annotated by linguists with the attributes necessary to train and evaluate the HSBN. The attributes annotated for each

Signer ID	Number of productions from the ASLLVD		
	HSBN training set	HSBN retrieval set (subset of training set)	HSBN test set
F1	2,567	728	–
M1	–	–	646
F2	1,162	688	–
M2	2,515	–	–
F3	381	–	–
F4	333	–	–
All	6,958	1,416	646

**Table 9.1:** Statistics for the productions of monomorphemic lexical signs from six native signers contained in the HSBN training, retrieval and test sets. The sizes of the retrieval and test sets are constrained by the availability of bounding box annotations for the start/end handshapes.

	HSBN training set	HSBN retrieval set	HSBN test set
# distinct signs	2,636	783	577

**Table 9.2:** Statistics for the number of distinct monomorphemic lexical signs in the HSBN training, retrieval and test sets.

Articulatory sub-class of monomorphemic lexical signs	Number of productions from the ASLLVD		
	HSBN training set	HSBN retrieval set	HSBN test set
one-handed	2,258	408	176
two-handed : same handshapes	3,072	670	320
two-handed : different handshapes	1,629	338	150

**Table 9.3:** Statistics for the different articulatory classes contained in the HSBN training, retrieval and test sets.

monomorphemic lexical sign include the locations of the start/end frames, an articulatory class label (one-handed, two-handed : same handshapes, or, two-handed : different handshapes) for each sign, start/end handshape labels on the dominant hand in one-handed signs and

on both hands in two-handed signs, and, a gloss label that uniquely identifies each sign with a specific item in the vocabulary. Multiple productions from native sign language users (two male and four female signers) are available in a majority of signs as summarized in Chapter 4. The availability of multiple examples for several signs is essential for the HSBN learning algorithm to accrue sign independent, and also signer-independent, patterns of handshape variation.

The primary inputs required for learning the HSBN are start/end handshape labels of signs in a training set, the articulatory class label for these signs and a grouping of signs into distinct lexical items. A database of start/end hand images (which we term as the ‘retrieval set’) annotated with handshape labels is required in order to compute the observation likelihoods for handshapes observed in the query sign. The algorithm for start/end handshape inference requires the articulatory class associated with a query sign, the start/end video frames and the start/end hand location bounding boxes. Ground-truth handshape labels for signs in the test set are also needed in order to evaluate the handshape inference accuracy.

The test and training and sets for HSBN evaluation are obtained by partitioning the set of monomorphemic lexical signs contained in the ASLLVD. Towards our objectives of assessing handshape inference performance in a person-independent recognition scenario, the signs in the test and training sets were obtained from different signers. The retrieval set is constructed from a subset of signs in the training set. These three datasets were prepared as follows.

We identified one of the six signers (M1) as the test-user. The test set consists of the subset of signs obtained from M1 where we have hand location annotations. We restrict our attention to one specific test-user due to the computational expense involved in performing image alignment based handshape retrieval for computing the observation likelihoods during handshape inference. Signs in the retrieval set are a subset of the signs in the training set from signers F1 and F2 with start/end hand location annotations. The setup employed here for evaluating the HSBN is challenging because the retrieval set consists of start/end

handshapes in signs produced by female signers whereas the test set consists of handshapes in signs produced by a male signer. The numbers of productions from different signers contained in the three datasets are summarized in Table 9.3. The three columns list the number of productions of signs in the training set, the retrieval set and the test set. The numbers of distinct monomorphemic lexical signs that correspond to these productions are given in Table 9.2.

In preparing the database of handshape images for handshape retrieval, we use the natural distribution of handshapes as observed in the retrieval set. The expressions for handshape inference derived in Equations 5.2 and 5.4 include the necessary normalizing terms in the denominator for the start/end handshape frequency distributions. Handshapes on the non-dominant hand were included after mirroring about the vertical axis. The retrieval set contains 5226 handshape images.

The distributions of signs belonging to the different articulatory classes in the training, retrieval and test sets are summarized in Table 9.3. For the evaluation conducted here we did not implement the training and inference algorithms for the  $\text{HSBN}^{\text{non-dominant}}$  model proposed in Section 5.2 to represent the properties specifically ascribed to handshapes articulated on the non-dominant hand in `two-handed : different handshapes` signs. Signs in the latter class were therefore grouped together with `one-handed` signs (handshapes on the non-dominant hand for these signs were not considered during training and inference, they were however included in the retrieval set). A subset of `two-handed : same handshapes` signs in the ASLLVD was classified by linguists as ‘`two-handed : same handshapes : alternating movement`’ signs. The articulation of the two hands in these signs are out of phase by  $\approx 180$  degrees. Examples include signs where one hand translates forwards into the signing space while the other hand moves towards the signer, or, where one hand exhibits an open-to-close change in the hand configuration while the other hand exhibits a close-to-open change in hand configuration. A large fraction of signs in this class does not exhibit a change in handshape and these signs were pooled together with `two-handed : same handshapes` signs for the handshape inference experiments (323 productions of such signs are present in the ASLLVD).

In ‘two-handed : same handshapes : alternating movement’ signs with different start and end handshapes, however, the start (and end) points for the basic movements on the two hands differ. Since our annotations included identification of only a single start and end point for the two hands of the sign as a whole, we have removed these signs (31 productions in the ASLLVD) from consideration in this research.

## 9.2 Learning the HSBN

The HSBNStateSpaceEstimation (Algorithm 7.1) is used in training the HSBN. The HSBNStateSpaceEstimation algorithm maximizes the variational Bayes lower bound,  $L_{\mathbf{Z}}^{\text{VB}}$ , with respect to the hidden variable state-space,  $\mathbf{Z}$ , and the model hyper-parameters,  $\omega_{\mathbf{Z}}$ . In order to perform this maximization, the HSBNStateSpaceEstimation algorithm invokes the VBEM algorithm (Algorithm 6.4) several times to generate the state-space hypotheses that are required in each learning epoch. The latter algorithm was therefore implemented in optimized multi-threaded C code while the other components of the HSBNStateSpaceEstimation algorithm were implemented in Matlab. The HSBNStateSpaceEstimation algorithm requires approximately three days utilizing 25 threads on a 32 core Intel Xeon E5-2680 compute node to complete 200 learning epochs. The memory requirements are nominal, however:  $\approx 5\text{GB}$  for the current training set size.

The main inputs required for the HSBNStateSpaceEstimation algorithm are the training set,  $\mathbf{x}$ , and the hyper-parameters for priors,  $\omega_{\tau=1}^{\text{prior}}$ , associated with the model parameters in the first epoch. The training set was prepared as described in Section 9.1. The specific implementation chosen here to construct the hyper-parameters for priors is described in the next section.

### 9.2.1 Hyper-parameters for prior distributions

During initialization of the HSBNStateSpaceEstimation algorithm in the first learning epoch (the corresponding algorithm is listed in Algorithm 7.2), a one-to-one correspondence is assumed between the hidden states,  $\mathcal{Z}_{\tau=1}^s, \mathcal{Z}_{\tau=1}^e$ , and the set of handshape labels,  $\mathcal{X}$ . This assumed correspondence also facilitates in choosing appropriate hyper-parameters for priors

in the first learning epoch,  $\omega_{\tau=1}^{\text{prior}} = \{\nu_{\tau=1}^{\text{prior}}, \alpha_{\tau=1}^{\text{prior}}, \beta_{\tau=1}^{\text{s prior}}, \beta_{\tau=1}^{\text{e prior}}\}$ . These hyper-parameters represent Dirichlet distributions for priors defined over the multinomial model parameters,  $\lambda_{\tau=1} = \{\pi_{\tau=1}, \mathbf{a}_{\tau=1}, \mathbf{b}_{\tau=1}^{\text{s}}, \mathbf{b}_{\tau=1}^{\text{e}}\}$ . The latter are respectively the parameters of the probability distribution for start latent states, the probability distributions for start  $\rightarrow$  end transitions, and, the probability distributions for start and end latent state emissions. The model parameters and their associated hyper-parameters are also summarized in Tables 5.2 and 6.2.

We use the illustration depicted in Figure 9.1 to highlight a few of the different choices (among several others) that are available when defining hyper-parameters for Dirichlet priors. For this illustration we chose priors for the emission distributions,  $\beta_{\tau=1}^{\text{prior}}$ . The flat prior shown in Figure 9.1(a) does not influence parameter estimation, the learning in this case is therefore purely data-driven. The uniform hyper-priors in Figure 9.1(b) bias the multinomial parameters towards assuming equal values. Larger values for hyper-parameters increase this bias. The uniform prior therefore serves to reduce the spread of values for the estimated multinomial parameters. The diagonal dominant prior in Figure 9.1(c) encodes the notion that each latent state is primarily responsible for emitting its corresponding handshape label. The above three classes of priors are data-agnostic. A prior of the form displayed in Figure 9.1(d) can be useful for the purposes of encapsulating certain domain knowledge derived properties that are specific to a given learning task. The latter version was therefore adopted for the hyper-parameters of priors associated with the transition and emission distributions.

We restricted our attention to hyper-parameter values  $\beta_{\tau=1}^{\text{prior}}[z_i, x_j] \geq 1$  because the corresponding prior distribution has a simple interpretation in terms of synthetic examples having been hypothesized for each location  $(z_i, x_j)$  in the array with a frequency equal to  $\beta_{\tau=1}^{\text{prior}}[z_i, x_j] - 1$ . We refer to the set of coordinates in the hyper-parameter array with values  $\beta_{\tau=1}^{\text{prior}}[z_i, x_j] > 1$  as ‘selected’ locations in the following discussion.

The hyper-parameters,  $\nu_{\tau=1}^{\text{prior}}$ , were chosen to specify a flat prior distribution for the model parameters,  $\pi_{\tau=1}$ .

The hyper-parameters,  $\alpha_{\tau=1}^{\text{prior}}$ , are derived from the matrix of start $\rightarrow$ end handshape



$\beta_{\tau=1}^{\text{prior}}[z_i, x_j]$	$x_1$	$x_2$	$x_3$
$z_1$	1	1	1
$z_2$	1	1	1
$z_3$	1	1	1

(a) ‘Flat’ prior

$\beta_{\tau=1}^{\text{prior}}[z_i, x_j]$	$x_1$	$x_2$	$x_3$
$z_1$	<b>2</b>	<b>2</b>	<b>2</b>
$z_2$	<b>2</b>	<b>2</b>	<b>2</b>
$z_3$	<b>2</b>	<b>2</b>	<b>2</b>

$\beta_{\tau=1}^{\text{prior}}[z_i, x_j]$	$x_1$	$x_2$	$x_3$
$z_1$	<b>5</b>	<b>5</b>	<b>5</b>
$z_2$	<b>5</b>	<b>5</b>	<b>5</b>
$z_3$	<b>5</b>	<b>5</b>	<b>5</b>

(b) ‘Uniform’ prior with lower (left) and higher (right) degrees of influence in drawing the estimated multinomial parameters towards the uniform distribution

$\beta_{\tau=1}^{\text{prior}}[z_i, x_j]$	$x_1$	$x_2$	$x_3$
$z_1$	<b>5</b>	1	1
$z_2$	1	<b>5</b>	1
$z_3$	1	1	<b>5</b>

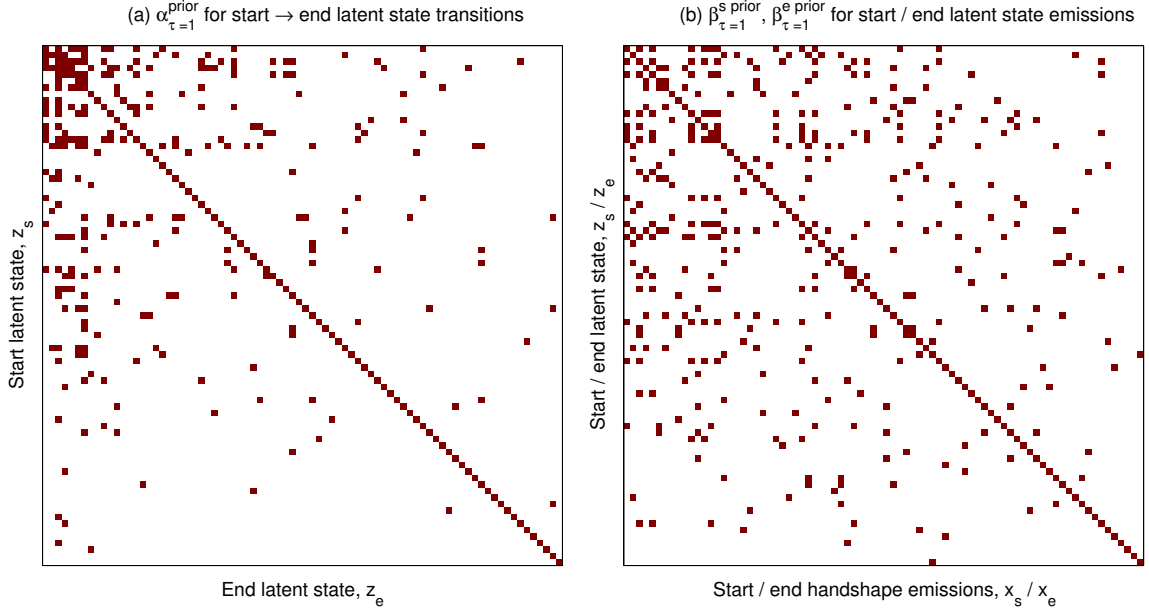
$\beta_{\tau=1}^{\text{prior}}[z_i, x_j]$	$x_1$	$x_2$	$x_3$
$z_1$	<b>5</b>	<b>5</b>	1
$z_2$	1	<b>5</b>	<b>5</b>
$z_3$	<b>5</b>	1	<b>5</b>

(c) ‘Diagonal dominant’ prior

(d) ‘Domain dependent’ prior

**Figure 9-1:** An illustration of a few different choices (among several others) that are possible in defining the hyper-parameters,  $\beta_{\tau=1}^{\text{prior}}$ , of Dirichlet priors in the first learning epoch for the multinomial emission distribution parameters,  $\mathbf{b}_{\tau=1}$ . Here,  $z \in Z$  are the latent states and  $x \in \mathcal{X}$  are the observed handshape labels. Hyper-parameter values  $\geq 1$  were chosen because the corresponding Dirichlet distributions have easy to interpret properties (the actual numerical values are chosen for illustration).

transition frequencies for signs in the training set. Examples for start/end handshape co-occurrences are shown in Table 9.4. The first image in each row of the figure shows a handshape, along with the number of instances where it appears in the ASLLVD. The handshape B-L, for example, appears 1162 times in the dataset. The remaining handshapes in a given row of the figure correspond to handshapes that occur as the end handshape.



**Figure 9-2:** The hyper-parameters,  $\alpha_{\tau=1}^{\text{prior}}$ ,  $\beta_{\tau=1}^{\text{s prior}}$ ,  $\beta_{\tau=1}^{\text{e prior}}$ , specified for prior distributions associated with the model parameters  $\mathbf{a}$ ,  $\mathbf{b}^{\text{s}}$ ,  $\mathbf{b}^{\text{e}}$  are displayed above. The selected locations in the above arrays are set to the value 3 while the unselected locations are set to the value 1.

Locations with non-zero values in the start/end handshake co-occurrence matrix are chosen as the selected locations for the  $\alpha_{\tau=1}^{\text{prior}}$  hyper-parameter array. The hyper-parameters for the transition distributions in the first learning epoch are displayed in Figure 9-2(a). The number of rows and columns in this array corresponds to the number of handshake labels,  $|\mathcal{X}|$ . The selected locations in the hyper-parameter arrays were set to 3 while the unselected locations were set to 1.

The hyper-parameters,  $\beta_{\tau=1}^{\text{s prior}}$ ,  $\beta_{\tau=1}^{\text{e prior}}$ , associated with the start/end observation probability distribution parameters,  $\mathbf{b}_{\tau=1}^{\text{s}}$ ,  $\mathbf{b}_{\tau=1}^{\text{e}}$ , play an important role in the learning. These are derived from the statistics of handshakes that were observed to have been produced in variation with other handshakes among multiple productions of the same lexical item in the training set. The computation of these statistics is described first followed by the procedure adopted for constructing the hyper-parameters,  $\beta_{\tau=1}^{\text{s prior}}$ ,  $\beta_{\tau=1}^{\text{e prior}}$ .

Start/end handshake labels from different productions of lexical items in the ASLLVD

are used to construct the table of handshapes that are observed to have been produced together with other handshapes. Examples from this table are shown in Table 9.5. The first image in each row of the figure shows a handshape, along with the number of instances where it appears in the ASLLVD. The remaining handshapes in a given row of the figure correspond to handshapes that appear in other instances of the same sign in the ASLLVD (each instance corresponds to one video clip of a monomorphemic lexical sign in the ASLLVD lexicon). The statistics of handshapes that were observed to have been produced together are displayed using ratios in each cell of the above table. Taking as an example the cell in the top-row, second column of this table, we observe that among the set of lexical items from the vocabulary where the handshapes B-L and B have been produced, there were 240 instances that were annotated with the handshape label B-L while there were 160 other instances that were annotated with the handshape label B. The number of lexical items from among which the above ratios were obtained for each cell has not been included in this table.

In preparing the above table, to ensure equivalence with the modeling assumptions made in the HSBN<sup>congruent</sup> representation, we did not distinguish between whether the handshape appears on the dominant hand or on the non-dominant hand in **two-handed : same handshapes** signs. However, handshape variants that occurred in the start and end positions of signs were considered separately and their statistics were later accumulated together to produce the counts in the above table. It is an empirical question whether the handshapes found at the start and end points of signs exhibit the same (or perhaps slightly different) types of variation. In designing the learning algorithm for the HSBN, we chose to separate the start and end latent states and their emission distribution parameters (the training data therefore determines the respective properties). A common set of hyper-parameters for the priors were however specified for both the start and end handshape emission distributions.

Several factors contribute to variations in articulation among signs produced by native sign language users [Battison et al., 1975, Van der Kooij, 2002, Bayley et al., 2002]. In formulating the HSBN, we restricted our attention to start/end handshape variations in

monomorphemic lexical signs. The algorithm proposed for learning the HSBN relies on start/end handshape labels annotated for productions of signs in the training set and also relies on the grouping of productions into different lexical items. The latter groupings are necessary for the proposed algorithm to learn the patterns of phonological handshape variations (i.e., sign- and signer-independent handshape variations) that occur in different productions of the same lexical item. Preparing the annotations for both these attributes (along with many other linguistic properties) presented several daunting challenges, however. The difficulties we faced in annotating a large collection ( $\approx 10,000$ ) of signs resulted in some handshape combinations appearing in this set that do not, in fact, truly represent phonological variants. Ensuring consistency among annotations provided by a large number of student annotators was extremely difficult, particularly since we had available a discrete set of handshape labels to account for handshapes that frequently did not exactly match any of our labelled handshapes, but, for example, exhibited properties that were intermediate between two different handshapes (e.g., with respect to degree of curvature of the fingers, or the degree to which they were spread). Parts of the hands for many signs are often occluded in both front and side views and there is hence some degree of uncertainty in some of the annotated handshape labels. Furthermore, annotators (and also the signers) may have differed slightly in assessing the exact start point of a sign in which the hand configuration changes over the course of production of that sign. To maximize the degree of consistency in the annotations, several passes of verifications were made, but there are surely still cases where the differences in the annotations of two different signs would suggest a greater degree of difference between the actual productions than is actually attested. Despite the painstaking efforts of linguists, the determination of whether two productions should be considered to be instances of a single lexical item was also not totally straightforward. This again required judgment calls about degrees of difference in articulation and meaning. The difficulties involved in such categorizations also contributes to confounds in the sets of apparent handshape variants.















































































A carefully specified set of hyper-parameters,  $\beta_{\tau=1}^s$  prior,  $\beta_{\tau=1}^e$  prior, for the prior distributions

provides one means to ensure that,  $\beta_{\tau}^{s*}$ ,  $\beta_{\tau}^{e*}$ , the hyper-parameters of emission distributions for latent states inferred by the learning algorithm are appropriate for the class of phonological variations that are representable in the HSBN. To construct the hyper-parameters for priors, we start with the table of handshape variations (as in Table 9.5) whose statistics were computed from signs in the training set. From among the handshape labels that were attested as having been produced in free variation with the handshape shown in the first column, we separated out handshape labels that were regarded as particularly unlikely to arise as phonological variants of the handshape label in the first column. Thus the set of handshape variants in each row was split into a ‘primary’ set and a ‘secondary’ set (Table 9.6 displays examples of annotations of these separations for first six handshapes from Table 9.5). The determination of these subsets was performed by the author based on the perceived similarity in internal configuration among handshape pairs. These selections do not carry any particular linguistic significance, however. An extensive linguistic analysis with a substantially larger dataset and many more signers is necessary to obtain deeper insights into the phenomenon of phonological variation in sign language. An alternate method for specifying the hyper-parameters of the priors could be derived based on comparing the 3D internal configurations for the handshapes.

The hyper-parameters,  $\beta_{\tau=1}^{s \text{ prior}}$ ,  $\beta_{\tau=1}^{e \text{ prior}}$ , of the start/end emission distribution priors specified in the initial learning epoch share the same values. The corresponding hyper-parameter matrix is displayed in Figure 9.2(b). The number of rows and columns in this matrix corresponds to the number of handshape labels,  $|\mathcal{X}|$ . The locations in the hyper-parameter arrays that were annotated as ‘primary handshape variations’ were set to 3 while the remaining locations were set to 1. The hyper-parameter array is nearly symmetric about the main diagonal because it reflects the symmetry properties of the table of handshapes that were observed to have been produced together.

Handshapes that occur as END handshapes in signs where the handshape in the first column occurs as the START handshape																	
B-L 1162	B-L 1007	bent-B-L 84	10 24	flat-O 16	crvd-B 16	A 6	5 2	crvd-5 2	Y 2	crvd-sprd-B 1	bent-B 1	tight-C 1					
1 976	1 864	X 62	bent-1 27	5 8	S 5	cocked-S 4	A 3	X-over-thumb 2	B-L 1								
5 893	5 613	S 101	flat-O 48	5-C 33	A 25	crvd-5 23	8 15	10 9	crvd-sprd-B 8	fanned-flat-O 6	B-L 4	5-C-L 3	crvd-B 1	4 1	bent-B 1	O 1	5-C-tt 1
S 766	S 531	5 103	crvd-5 43	1 32	4 10	V/2 9	U/H 9	crvd-sprd-B 8	C 4	bent-1 4	3 4	B-L 2	bent-B-L 2	P/K 2	W 2	crvd-W 1	
crvd-5 567	crvd-5 389	S 83	flat-O 30	A 14	5-C 14	5-C-L 7	10 6	bent-B-L 6	crvd-sprd-B 5	8 4	5 3	crvd-B 3	tight-C 2	5-C-tt 1			
10 512	10 506	A 4	5 2														

**Table 9.4:** Table of start/end handshape co-occurrences computed from handshapes on the dominant hand in one-handed and two-handed : different handshapes signs, and, from handshapes on the dominant and non-dominant hands in two-handed : same handshapes signs. Monomorphemic signs contained in the ASLLVD were used in preparing this table.

Handshapes produced in variation with the handshape shown in the first column among multiple productions of the same lexical item												
B-L 1157 	B 240 / 160 	flat-B 235 / 98 	bent-B-L 218 / 146 	5 178 / 201 	crvd-B 151 / 95 	B-xd 68 / 50 	bent-B 65 / 34 	crvd-flat-B 28 / 18 	crvd-5 24 / 19 	crvd-sprd-B 21 / 13 	S 10 / 2 	C 6 / 6 
1 975 	bent-1 190 / 105 	D 78 / 15 	L 31 / 13 	X 31 / 37 	cocked-S 11 / 4 	alt-P 8 / 4 	5 6 / 22 	G/Q 6 / 6 	alt-G 4 / 2 	L-X 2 / 2 	A 1 / 1 	X-over-thumb 1 / 1 
5 891 	crvd-5 300 / 227 	B-L 201 / 178 	crvd-sprd-B 79 / 28 	4 47 / 53 	5-C 23 / 10 	1 22 / 6 	crvd-B 21 / 27 	25 21 / 19 	5-C-L 10 / 9 	bent-B-L 9 / 4 	fanned-flat-O 9 / 6 	open-8 7 / 15 
S 757 	A 249 / 172 	10 31 / 34 	X-over-thumb 12 / 32 	flat-O 11 / 14 	crvd-sprd-B 10 / 3 	cocked-U 9 / 15 	crvd-B 8 / 2 	cocked-S 7 / 6 	crvd-5 6 / 6 	baby-O 3 / 4 	B-L 2 / 10 	crvd-3 2 / 3 
crvd-5 561 	5 227 / 300 	5-C-L 116 / 58 	C 92 / 49 	5-C 92 / 50 	crvd-sprd-B 75 / 52 	crvd-B 72 / 76 	B-L 19 / 24 	A 10 / 2 	F/9 10 / 2 	flat-O 10 / 2 	bent-B-L 10 / 8 	loose-E 10 / 4 
10 512 	A 366 / 225 	S 34 / 31 	B-xd 10 / 2 	L-X 8 / 10 	Y 6 / 2 	Horns 6 / 4 	bent-Horns 6 / 8 	crvd-B 5 / 1 	X-over-thumb 5 / 4 	U-L 5 / 2 	bent-U 5 / 1 	bent-U-L 5 / 5 

**Table 9.5:** Start/end handshape labels from different productions of lexical items in the ASLLVD are used to construct the table of handshapes that are observed to have been produced in free variation with other handshapes. The ratios shown in each cell of the above table are computed as follows: among the subset of lexical items in the ASLLVD where the row and column handshape classes are produced together, the numerator counts the number of times the row shape class appears and denominator counts the number of times the column shape class appears.

Handshapes produced in variation with the handshape shown in the first column among multiple productions of the same lexical item:  
Variants that were annotated as "primary variants"

B-L 1157	B 240 / 160	flat-B 235 / 98	bent-B-L 218 / 146	S 178 / 201	crvd-B 151 / 95	B-xd 68 / 50	bent-B 65 / 34	crvd-flat-B 28 / 18	crvd-5 24 / 19	crvd-sprd-B 21 / 13	C 6 / 6
1 975	bent-1 190 / 105	D 78 / 15	L 31 / 13	X 31 / 37	alt-P 8 / 4	G/Q 6 / 6	alt-G 4 / 2	L-X 2 / 2			
5 891	crvd-5 300 / 227	B-L 201 / 178	crvd-sprd-B 79 / 28	4 47 / 53	S-C 23 / 10	crvd-B 21 / 27	25 21 / 19	S-C-L 10 / 9	bent-B-L 9 / 4	B 4 / 3	S-C-tt 4 / 2
S 757	A 249 / 172	10 31 / 34	X-over-thumb 12 / 32	flat-O 11 / 14	cocked-U 9 / 15	cocked-S 7 / 6	baby-O 3 / 4	O 1 / 1			
crvd-5 561	S 227 / 300	S-C-L 116 / 58	C 92 / 49	O 92 / 50	crvd-sprd-B 75 / 52	crvd-B 72 / 76	B-L 19 / 24	bent-B-L 10 / 8	loose-E 10 / 4	S-C-tt 6 / 6	
10 512	A 366 / 225	S 34 / 31	L-X 8 / 10	X-over-thumb 5 / 4							

Handshapes produced in variation with the handshape shown in the first column among multiple productions of the same lexical item:  
Variants that were annotated as "secondary variants"

B-L 1157	X S 10 / 2	X light-C 3 / 3										
1 975	X cocked-S 11 / 4	X S 6 / 22	X A 1 / 1	X X-over-thumb 1 / 1								
5 891	X 1 22 / 6	X fanned-flat-O 9 / 6	X open-B 7 / 15	X A 4 / 4	X open-F 4 / 7	X flat-O 3 / 12	X O 3 / 6	X F/9 2 / 2	X loose-E 2 / 4	X cocked-7 2 / 2	X crvd-W 1 / 1	X full-M 1 / 1
S 757	X crvd-sprd-B 10 / 3	X crvd-B 8 / 2	X crvd-5 6 / 6	X B-L 2 / 10	X crvd-3 2 / 3	X X 1 / 8	X bent-1 1 / 2	X L-X 1 / 2	X bent-U 1 / 1	X bent-U-L 1 / 5		
crvd-5 561	X A 10 / 2	X F/9 10 / 2	X flat-O 10 / 2	X S 6 / 6	X 25 2 / 4	X open-B 2 / 6	X X-over-thumb 1 / 2	X bent-B 1 / 1	X baby-O 1 / 2			
10 512	X B-xd 10 / 2	X Y 6 / 2	X Horns 6 / 4	X bent-Horns 6 / 8	X crvd-B 5 / 1	X U-L 5 / 2	X bent-U 5 / 1	X bent-U-L 5 / 5	X bent-B-L 4 / 4	X L 4 / 4	X light-C 2 / 4	

**Table 9.6:** Examples of cells from the handshape variants table (Table 9.5) that were annotated by the author as 'primary' and 'secondary' variants are displayed in the left and right tables respectively. Primary handshape variants are used to specify the set of selected locations in the hyper-parameter arrays of the emission distribution priors,  $\beta_{\tau=1}^{s \text{ prior}}$ ,  $\beta_{\tau=1}^{e \text{ prior}}$ ; these hyper-parameters are displayed in Figure 9.2.



### 9.2.2 State-space refinement using the `HSBNStateSpaceSelection` algorithm

Given the training set and the hyper-parameters for priors specified in the first epoch, the initial estimates for the HSBN state-space,  $(\mathcal{Z}_{\tau=1}^s, \mathcal{Z}_{\tau=1}^e)$ , and the model parameters,  $\omega_{\tau=1}$ , are obtained using the HSBN initialization algorithm (Algorithm 7.2).

The convergence tolerances in the VBEM algorithm for the VB lower bound,  $L^{\text{VB}}$ , and for the estimated hyper-parameters,  $\omega$ , were set to 1 and  $10^{-2}$  respectively. The maximum number of EM iterations was chosen to be 50 (less than 25 EM iterations were typically required for convergence).

In subsequent epochs, the `HSBNStateSpaceSelection` algorithm (Algorithm 7.3) generates state-space candidates by applying different state-space refinements to the current model parameters,  $(\omega_{\tau}, \omega_{\tau}^{\text{prior}}, \mathbf{Z}_{\tau})$ . The number of candidates generated by the state-space refinement methods, merge-states, drop-state, reset-state, and add-state, are given by  $\left(\frac{|\mathcal{Z}_{\tau}^s| |\mathcal{Z}_{\tau}^s - 1|}{2} + \frac{|\mathcal{Z}_{\tau}^e| |\mathcal{Z}_{\tau}^e - 1|}{2}\right)$ ,  $(|\mathcal{Z}_{\tau}^s| + |\mathcal{Z}_{\tau}^e|)$ ,  $(|\mathcal{Z}_{\tau}^s| + |\mathcal{Z}_{\tau}^e|)$ , and,  $(|\mathcal{Z}_{\tau}^s \setminus \mathcal{Z}_{\tau=1}^s| + |\mathcal{Z}_{\tau}^e \setminus \mathcal{Z}_{\tau=1}^e|)$  respectively. The `HSBNStateSpaceSelection` algorithm retains a fixed number of candidates ( $\leq 30$ ) with the largest estimated values of the VBEM lower bound for each refinement type. An acceptance ratio distribution for the retained candidates is computed as given in Algorithm 7.3, step 11. The state-space for the next epoch is sampled from the acceptance ratio distribution (Algorithm 7.3, steps 12-13).

The number of learning epochs for the `HSBNStateSpaceEstimation` algorithm was chosen to be 200 based on the fact that 85 states were selected for the start and end hidden variable state-spaces in the first learning epoch. This choice ensures that an adequate number of learning epochs were provided for the state-space optimization algorithm. At the conclusion of the `HSBNStateSpaceEstimation` optimization procedure, the estimated state-space,  $\hat{\mathbf{Z}} = (\hat{\mathbf{Z}}^s, \hat{\mathbf{Z}}^e)$ , and the corresponding hyper-parameters,  $\omega_{\hat{\mathbf{Z}}}$ , are obtained as outputs. A point estimate for the model parameters,  $\lambda_{\hat{\mathbf{Z}}}$ , is obtained by computing the expected values of the model parameter distributions with hyper-parameters,  $\omega_{\hat{\mathbf{Z}}}$  (Equation 6.48).

### 9.3 Handshape retrieval to compute observation likelihoods

The HSBN handshape inference algorithm utilizes simple nearest neighbor retrieval to compute observation likelihood scores for start/end hand images obtained from input video of a sign. The database of hand images for retrieval is constructed as described in Section 9.1. Given the retrieved list of handshape labels, the tunable parameters for computing the observation likelihood scores are those contained in Equation 8.1. These are,  $K$ , the number of examples retrieved during k-NN search and the exponential weighting parameter,  $\beta$ . The value for  $\beta$  was set to 0.1 and  $K$  was set to 200. The influence of  $\beta$  on handshape inference accuracy is analyzed further in the next chapter.

The following three methods for computing a similarity score for hand image pairs are chosen here to compare their k-NN handshape retrieval performance: ‘no image alignment’, ‘affine image alignment’ and ‘non-rigid image alignment’. All three algorithms employ the same feature representation but differ in the amounts of displacement allowed between feature locations in the two images. The similarity score in all three methods is computed using the data-association cost formulated in Equations 8.6 and 8.19.

#### 9.3.1 Pre-processing of hand images

The hand images were cropped and centered with respect to the handshapes observed in video. The bounding box annotations were restricted to a square aspect ratio. The hand images after cropping were normalized to  $90 \times 90$  pixels. We employ the steps described in [Thangali and Sclaroff, 2009] to pre-process the images obtained from sign language video sequences. The pre-processing steps include skin-color based image segmentation and subsequent morphological operations to clean-up the foreground segmentation results. The foreground/background classification algorithm utilizes RGB histograms trained using a subset of video frames from the ASLLVD collection annotated with foreground/background region information.

### 9.3.2 Local feature representation

Local image descriptors for handshape images were computed using the HOG method following the approach described in Section 8.2.2. The local orientations  $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_9\}$  for feature extraction at a feature location,  $\mathcal{P}_{x,y}^{\mathbf{j}}$ , in image  $\mathbf{j}$  are sampled uniformly in the range  $\left[\frac{-\pi}{10}, \frac{\pi}{10}\right]$ . The HOG feature descriptor  $\mathbf{h}_{x,y}^{\mathbf{j},\theta}$  for a specific  $\theta$  is computed as follows. A sub-image of size  $14 \times 14$  pixels centered at the feature location  $\mathcal{P}_{x,y}^{\mathbf{j}}$  is partitioned into non-overlapping  $2 \times 2$  blocks. Within each of these blocks the image gradient directions are aggregated into a normalized 9-bin orientation histogram. The resulting 36 dimensional feature vector normalized to a unit squared norm (to incorporate to some extent robustness to variations in imaging conditions) yields the required HOG feature descriptor,  $\mathbf{h}_{x,y}^{\mathbf{j},\theta}$ . Efficient online computation of the HOG features is accomplished by utilizing summed-area tables that are pre-computed for each of the 9 local image orientations and the 9 bin directions employed in the HOG representation. The handshape image is rotated into each of the different local orientations prior to computing the summed-area tables so as to ensure that we have axis-aligned rectangular regions when aggregating the contributions of per-pixel image gradients into each of the HOG orientation bins.

The ‘no image alignment’ method assumes one-to-one spatial correspondence between the feature locations  $\mathcal{G}_{k,l}^{\mathbf{i}}, \mathcal{G}_{k,l}^{\mathbf{j}}$  in the two images and therefore only requires a search over the different local orientations to compute the similarity score (as illustrated in Equation 8.5).

### 9.3.3 Computing the non-rigid image alignment

The parameters used in computing the bi-directional non-rigid image alignments for a handshape image pair using the proposed algorithm Algorithm 8.1 are summarized here. A control lattice consisting of  $12 \times 12$  equally spaced nodes is defined within the handshape image. An additional set of fixed nodes is included on the periphery of the handshape image. The structure of the spring mesh that connects the control lattice nodes is illustrated in Figure 8-1. Evaluating the similarity score for a pair of images,  $\text{sim}(\mathbf{i}, \mathbf{j})$ , involves computing the bi-directional image alignments  $\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}$  and  $\mathbf{a}^{\mathbf{j} \rightarrow \mathbf{i}}$ . The alignment vectors  $\mathbf{a}^{\mathbf{i} \rightarrow \mathbf{j}}$  map fea-

ture locations situated at the control lattice coordinates  $\mathcal{G}_{k,l}^i$  in image  $\mathbf{i}$  to pixel coordinates  $\mathcal{G}_{k,l}^j + \mathbf{a}_{k,l}^{\tau:i \rightarrow j}$  in image  $\mathbf{j}$  ( $\tau$  denotes the iteration index). The local neighborhood grid  $\mathcal{W}_{k,l}^{\tau:j}$  used in computing the local displacement vectors  $\Delta \mathbf{a}^{\tau:i \rightarrow j}$  in Equation 8.7 are specified with a 2 pixel spacing as follows:  $\mathcal{W}_{k,l}^{\tau:j} = \mathcal{G}_{k,l}^{\tau:j} + [-16 : 2 : 16] \otimes [-16 : 2 : 16]$ . The base spring stiffness parameter in Equation 8.15 was chosen to be 75. The spring lengths in Equation 8.9 are computed in pixel coordinates. The LSE in Equation 8.16 is solved using the conjugate gradient algorithm [Press et al., 2007]. A line-search to determine the optimal scaling for the alignment vectors in Equation 8.17 was performed using the golden section minimization algorithm [Press et al., 2007]. A value of 0.2 was chosen for the  $\alpha_{\max}$  parameter in Equation 8.17.

The computation of an affine alignment that serves as an initialization for the non-rigid image alignment algorithm utilizes the same sequence of steps as above. The parameters of the affine transformation matrix are computed using the least squares method.

### 9.3.4 Filter and refine handshape retrieval

Computing the non-rigid alignment for a hand image pair requires on average 2.04s while computing an affine alignment requires 0.66s. The similarity score computed using the latter method is therefore employed as a filtering step during handshape retrieval to shortlist candidates for subsequent refinement using the non-rigid alignment method. In our current implementation, the filtering step selects 1000 handshape candidates from among a total of 5226 hand images contained in the retrieval set. The MRF-LBP algorithm (Section 8.1) is another suitable candidate for comparison that was not included in our current evaluation due to its significantly higher computational cost (58s per image pair).

## 9.4 Handshape inference using the HSBN

Given the trained HSBN model parameters and the ranked lists of handshape labels retrieved from the database for the start/end handshapes in the query sign, we now describe the implementation of the algorithms to perform handshape inference.

We implemented algorithms for handshape inference using the `one-handed` and `two-handed:same` handshapes HSBN models. Both these algorithms use the same set of previously trained HSBN model parameters. The equations for handshape inference in `one-handed` signs are given in Equations 5.2 and 5.3. The respective equations for `two-handed:same` handshapes signs are given in Equations 5.4 and 5.5. In our current implementation the HSBN retrieval set is a small subset of signs from the HSBN training set; the frequencies of handshapes in these two sets therefore differ to some extent. To accommodate this difference, we utilize the frequencies for handshapes contained in the retrieval set for normalization terms in the denominators of Equations 5.2 and 5.4. The database constructed for handshape retrieval collects together hand images for start and end positions as well as for the dominant and non-dominant hands (the latter after flipping about the vertical axis). Therefore, a single handshape frequency distribution was used for all the denominator terms in the above two expressions for computing the posterior probabilities for different combinations of start/end handshapes.

In `one-handed` signs, the posterior probabilities  $P(X^{s:D} = x^{s:D}, X^{e:D} = x^{e:D} \mid I^{s:D} = \mathbf{i}^{s:D}, I^{e:D} = \mathbf{i}^{e:D})$  are computed for each of the different combinations of start/end handshape labels on the dominant hand,  $(X^{s:D} = x^{s:D}, X^{e:D} = x^{e:D})$ . In `two-handed:same` handshapes signs, the corresponding posterior distribution represents different combinations of start/end handshapes on both hands,  $(X^{s:D} = x^{s:D}, X^{e:D} = x^{e:D}, X^{s:N} = x^{s:N}, X^{e:N} = x^{e:N})$ . Arranging the handshape tuples in decreasing order of the estimated posterior probabilities produces the inferred list of start/end handshape tuples. The computations required for handshape inference can therefore be performed efficiently using closed form expressions in both cases.

## 9.5 Evaluating HSBN handshape inference performance

Ranked retrieval/inference accuracy was chosen here as the evaluation criterion. The ranked order of handshapes for simple-NN retrieval was obtained by retaining the first occurrence of each handshape label in the nearest neighbor list of retrieved handshapes. Simple-NN

was chosen as the baseline method to evaluate the HSBN’s handshake inference performance because this method does not involve tunable parameters ( $k$ , the size of the retrieved set influences the results only for large values of retrieved ranks). Other candidate baseline approaches such as a k-NN handshake classification method or the handshake ranking computed using the observation likelihood scores (Equation 8.1) were not compared here because the results produced by these methods are sensitive to the choice of the parameter  $k$  in both methods and the parameter  $\beta$  in the latter method.

We first take a `two-handed:same` handshapes query to illustrate the method used to compute the ranked orders of handshake labels from the HSBN handshake inference results. Given handshake images  $(\mathbf{i}_q^{s;D}, \mathbf{i}_q^{e;D}, \mathbf{i}_q^{s;N}, \mathbf{i}_q^{e;N})$  in a query sign (whose respective ground-truth handshake labels are  $(x_q^{s;D}, x_q^{e;D}, x_q^{s;N}, x_q^{e;N})$ ), the handshake inference algorithm yields a list of start/end handshake label tuples,  $(x_i^{s;D}, x_i^{e;D}, x_i^{s;N}, x_i^{e;N})$ ,  $1 \leq i \leq |\mathcal{X}|^4$ , arranged in decreasing order of their estimated joint posterior probabilities. The first occurrence of a handshake label in each of the following lists,  $\{x_i^{s;D}\}, \{x_i^{e;D}\}, \{x_i^{s;N}\}, \{x_i^{e;N}\}$ , is retained in order to produce the respective handshake label permutations,  $\{x_j^{s;D}\}, \{x_k^{e;D}\}, \{x_l^{s;N}\}, \{x_m^{e;N}\}$ ,  $1 \leq \{j, k, l, m\} \leq |\mathcal{X}|$ . The ground-truth handshake label’s position in the corresponding permutation list yields the handshake inference rank for the purposes of comparison with the respective simple-NN retrieved rank.

In a similar fashion, given handshake images  $(\mathbf{i}_q^{s;D}, \mathbf{i}_q^{e;D})$  in a `one-handed` query sign (whose respective ground-truth handshake labels are  $(x_q^{s;D}, x_q^{e;D})$ ), the handshake inference algorithm yields a list of start/end handshake label pairs,  $(x_i^{s;D}, x_i^{e;D})$ ,  $1 \leq i \leq |\mathcal{X}|^2$ , arranged in decreasing order of their estimated joint posterior probabilities. The first occurrence of a handshake label in each of the following lists,  $\{x_i^{s;D}\}, \{x_i^{e;D}\}$ , is retained in order to produce the respective handshake label permutations,  $\{x_j^{s;D}\}, \{x_k^{e;D}\}$ ,  $1 \leq \{j, k\} \leq |\mathcal{X}|$ . The ground-truth handshake label’s position in the corresponding permutation list, again, yields the handshake inference rank for the purposes of comparison with the respective simple-NN retrieved rank.

## Chapter 10

# Experiments: Results

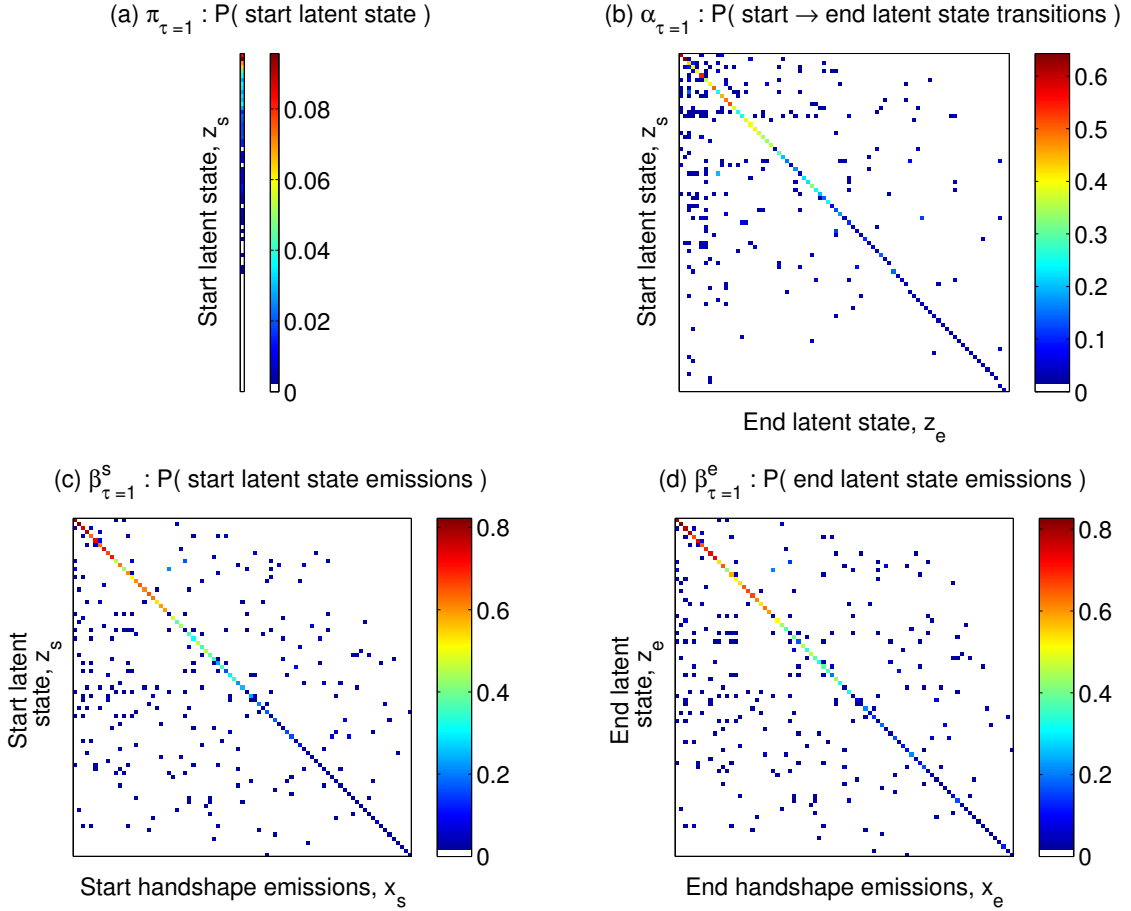
In this chapter we describe the results of the experiments that were conducted for learning the HSBN, for handshape retrieval and for handshape inference using the learnt model. The training, retrieval and test sets in these experiments were prepared as described in Chapter 9.

### 10.1 Learning the HSBN

We follow the implementation described in Section 9.2 to learn the HSBN. We present the results of initializing the model in the first learning epoch and describe the results obtained through the sequence of learning epochs for HSBN state-space refinement. To obtain additional insights, we compare the properties of the model estimated in the final learning epoch with those of the model prepared during initialization.

The hyper-parameters,  $\omega_{\tau=1}^*$ , produced by the VBEM algorithm in the first learning epoch are displayed in Figure 10-1. Among the signs contained in the HSBN training set certain handshapes were only observed to appear at either the start or the end positions, i.e.,  $\mathcal{X}^s, \mathcal{X}^e \subset \mathcal{X}$ . The HSBN initialization was performed assuming a one-to-one correspondence between the latent states  $\mathcal{Z}_{\tau=1}^s, \mathcal{Z}_{\tau=1}^e$  and the sets of observed handshape labels,  $\mathcal{X}^s, \mathcal{X}^e$ . Therefore,  $|\mathcal{Z}_{\tau=1}^s| \neq |\mathcal{Z}_{\tau=1}^e|$ , and this produces a discontinuity in the main diagonal of the hyper-parameter array for transitions,  $\alpha_{\tau=1}^*$ . (The same property also holds true for the hyper-parameters of the state transitions prior,  $\alpha_{\tau=1}^{\text{prior}}$ . For clarity of presentation, the hyper-parameter array shown in Figure 9-2(a) was reduced to a square matrix by leaving out the blank rows/columns.)

The columns that correspond to the observed handshape labels,  $\mathcal{X}^s, \mathcal{X}^e$ , were retained when rendering the emission distribution hyper-parameter arrays,  $\beta_{\tau=1}^{s*}, \beta_{\tau=1}^{e*}$  in Fig-

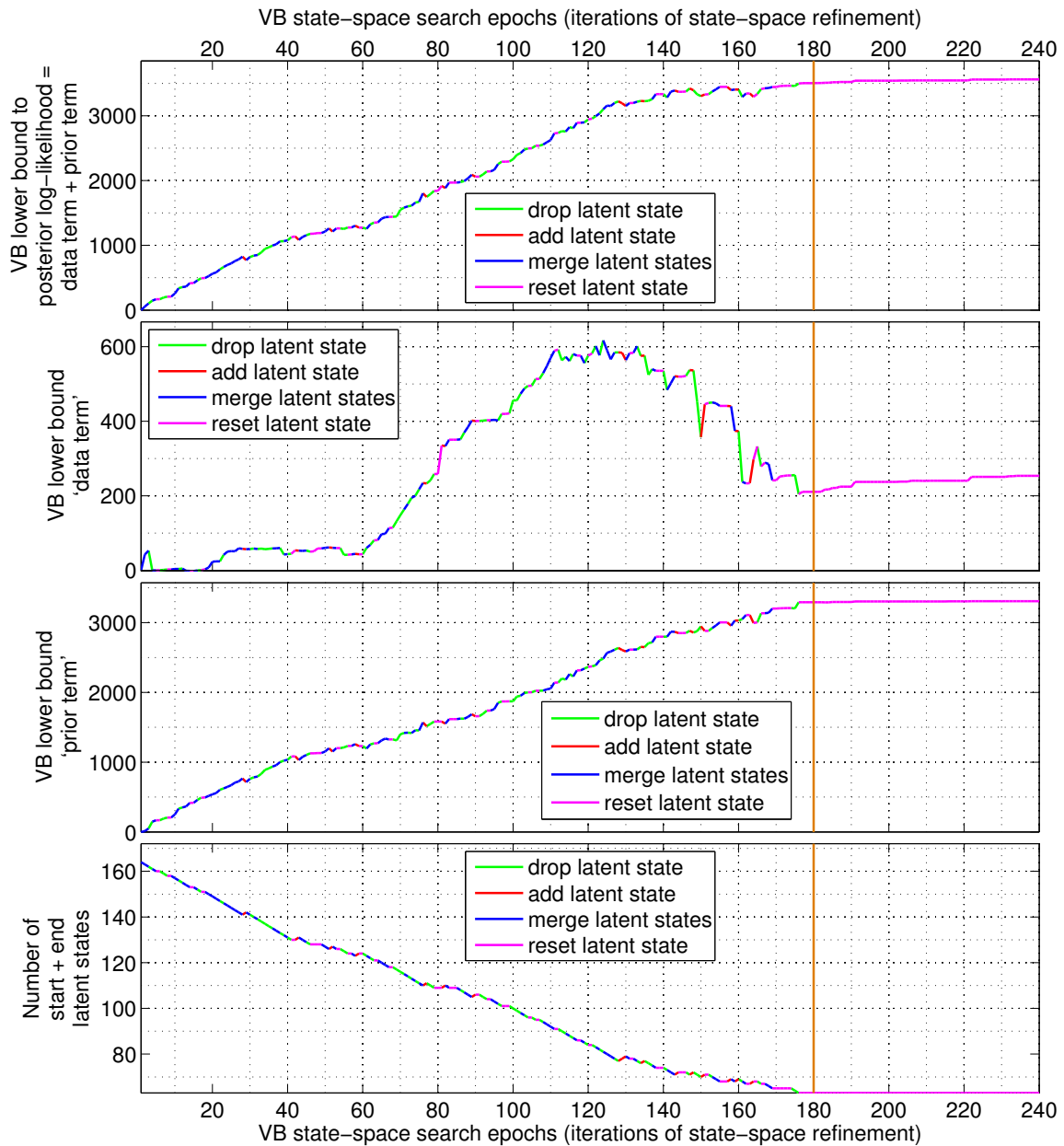


**Figure 10.1:** The normalized values for the hyper-parameters,  $\nu_{\tau=1}^*$ ,  $\alpha_{\tau=1}^*$ ,  $\beta_{\tau=1}^{s*}$ ,  $\beta_{\tau=1}^{e*}$ , estimated in the first learning epoch using the VBEM algorithm are displayed above. The above arrays also correspond to the estimated values for the HSBN multinomial parameters,  $\pi_{\tau=1}^*$ ,  $\mathbf{a}_{\tau=1}^*$ ,  $\mathbf{b}_{\tau=1}^{s*}$ ,  $\mathbf{b}_{\tau=1}^{e*}$ . The color bars depict values in the range  $[0, 1]$ .

ure 10.1. These two arrays are therefore square with no discontinuity in the values on the main diagonal.

The results of performing a sequence of state-space refinements using the HSBNStateSpaceEstimation algorithm are summarized in Figure 10.2. The top plot displays the values estimated for the VBEM lower bound,  $L_{\tau}^{\text{VB}}$ , in the sequence of learning epochs. Each of the four different colors in the two plots identifies the state-space refinement method selected by the HSBNStateSpaceSelection algorithm in a given learning epoch. For clarity, we do not





**Figure 10-2:** (a) The top plot displays the values of the estimated VBEM lower bound produced by the sequence of state-space refinements in the HSBNStateSpaceEstimation algorithm. (b,c) The VBEM lower bound sums together the contributions from the data-likelihood and the prior terms (the latter corresponds to the KL divergence between the probability distributions specified by the priors and the current model parameters). These two terms are shown in the second and third plots. The values from the first epoch have been subtracted out in the top three plots. (d) The bottom plot displays the evolution of the total number of latent states through the learning epochs.

distinguish between whether the state-space refinement selected by the algorithm applies to the start, or, to the end latent states. The selection of a state-space from among the generated candidates is performed in a stochastic fashion to aid in circumventing local maxima. The estimated value of the VBEM lower bound therefore decreases in some learning epochs.

The VBEM lower bound derived in Equation 6.59 sums together the contributions from the training data log-likelihood and the prior terms. These two components are graphed as a function of the learning epochs in the second and third plots. The values estimated in the first epoch have been subtracted out in the first three plots since only the relative values have a bearing on the learning algorithm. The data log-likelihood term appears as a sum over the normalizing constants,  $C_{Q_{\mathbf{z},i}}$ , for the variational distributions,  $Q_{\mathbf{z},i}$ , associated with each lexical item  $i$  in the training set vocabulary,  $\mathcal{V}_{\mathbf{x}}$ . The prior term appears as the KL divergence between the probability distributions represented by hyper-parameters of the prior,  $\omega_{\tau}^{\text{prior}}$ , and the hyper-parameters for the current model parameters,  $\omega_{\tau}$ .

The total numbers of latent states,  $|\mathcal{Z}_{\tau}^s| + |\mathcal{Z}_{\tau}^e|$ , estimated in the sequence of learning epochs are displayed in the last plot. The ‘merge-states’ and ‘drop-states’ refinement methods decrease the total number of latent states by 1, the ‘reset-state’ retains the current number of latent states and the ‘add-state’ refinement increases the number of latent states by 1.

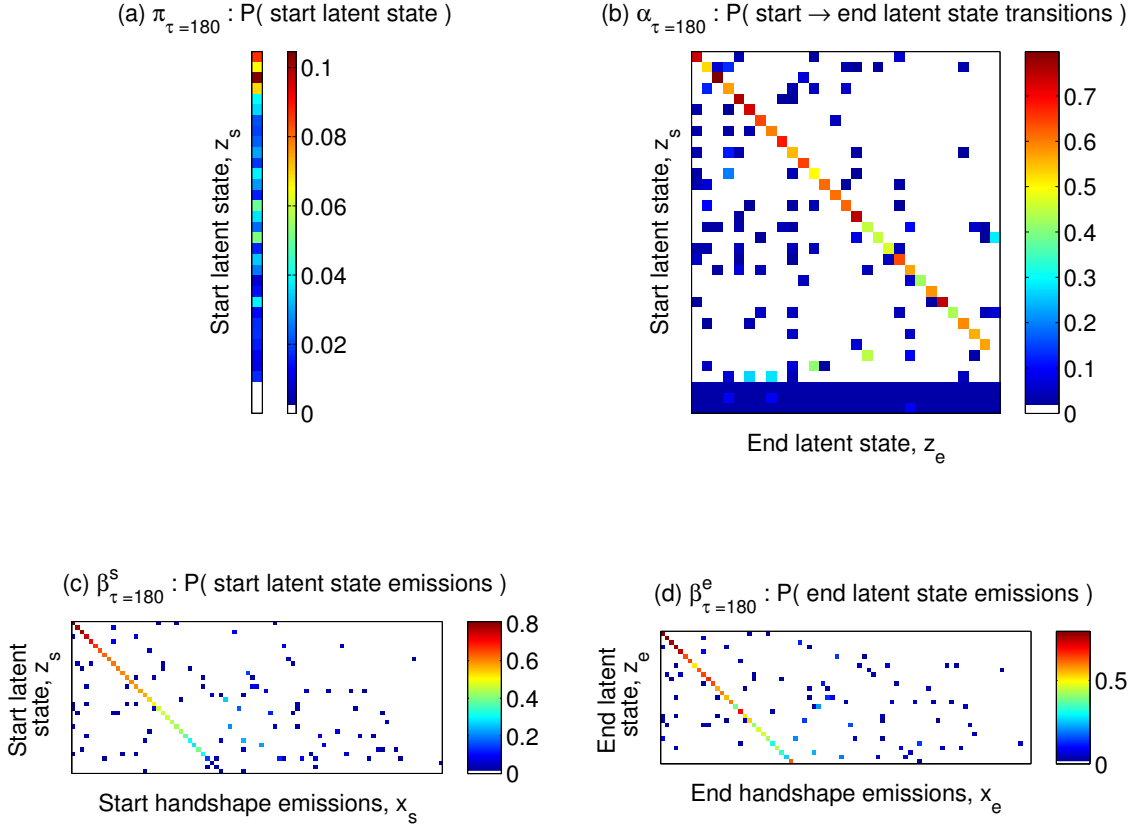
We surmise the following properties of the state-space estimation algorithm. Performing the sequence of state-space refinement steps with a goal towards increasing the VB lower bound reduces the total number of latent states employed in the HSBN model. The contribution from the prior term to the VB lower bound increases steadily through the learning epochs suggesting that the estimated model hyper-parameters are evolving towards the prior hyper-parameters. The data log-likelihood component of the VB lower bound increases up until the epoch 124, even though the latter model employs substantially fewer latent states than the initial model. This suggests that the model initialization selected for the first epoch in the VBEM optimization was sub-optimal (we may recall that the VBEM algorithm is a gradient ascent based method and is therefore sensitive to initialization).

The state-space refinement method, in order to generate state-space candidates for the next epoch, initializes the VBEM algorithm in several different ways and therefore is able to discover an initialization that improves both the VB lower bound as well as the training data log-likelihood. After a certain number of learning epochs, however, the prior term begins to dominate the optimization and the training data log-likelihood drops substantially. The choice of an appropriate epoch to terminate the learning is therefore crucial to ensure generalization performance of the estimated model.

A sequestered validation set is typically employed in learning approaches to select an epoch for early termination to circumvent, to some extent, the problems of over-fitting to a given training set. Since a validation set was not currently available in learning the HSBN, the final learning epoch,  $\tau = 180$ , was chosen by the author to balance the two components of the VB lower bound. A similar behavior was also observed in the other learning trials that we conducted.

The hyper-parameters,  $\omega_{\tau=180}^*$ , produced by the HSBNStateSpaceEstimation algorithm in the final learning epoch are displayed in Figure 10-1. The number of rows of  $\nu_{\tau=180}^*$ ,  $\alpha_{\tau=180}^*$ ,  $\beta_{\tau=180}^{s*}$  corresponds to the estimated number of start latent states,  $|\mathcal{Z}_{\tau=180}^s| = 34$ . The number of columns of  $\alpha_{\tau=180}^*$  and the number of rows of  $\beta_{\tau=180}^{e*}$  corresponds to the estimated number of end latent states  $|\mathcal{Z}_{\tau=180}^e| = 29$ . The columns of  $\beta_{\tau=180}^{s*}$ ,  $\beta_{\tau=180}^{e*}$  correspond to the observed handshape labels,  $\mathcal{X}^s$ ,  $\mathcal{X}^e$ .

To obtain further insights into the results produced by the HSBNStateSpaceEstimation algorithm, we visualize the hyper-parameters,  $\beta_{\tau=180}^{s*}$ ,  $\beta_{\tau=180}^{e*}$ , of the emission distributions for the start and end latent states estimated in the final learning epoch in Figures 10-4, 10-6. The HSBNStateSpaceEstimation algorithm was initialized with a set of,  $|\mathcal{Z}_{\tau=1}^s| = 83$ , start latent states and  $|\mathcal{Z}_{\tau=1}^e| = 81$ , end latent states. Through the process of maximizing the VB lower bound the HSBN learning algorithm arrives at a set of,  $|\mathcal{Z}_{\tau=180}^s| = 34$ , start latent states and,  $|\mathcal{Z}_{\tau=180}^e| = 29$ , end latent states. In order to convey the properties of latent states estimated by the proposed learning algorithm, we display the list of handshape labels,  $x_i$ , that are associated with each end latent state in decreasing order of the estimated



**Figure 10-3:** The normalized values for the hyper-parameters,  $\nu_{\tau=180}^*$ ,  $\alpha_{\tau=180}^*$ ,  $\beta_{\tau=180}^{s*}$ ,  $\beta_{\tau=180}^{e*}$ , that were estimated in the final learning epoch of the HSBNStateSpaceEstimation algorithm are displayed above. The above arrays also correspond to the estimated values for the HSBN multinomial parameters,  $\pi_{\tau=180}^*$ ,  $\mathbf{a}_{\tau=180}^*$ ,  $\mathbf{b}_{\tau=180}^{s*}$ ,  $\mathbf{b}_{\tau=180}^{e*}$ . The color bars depict values in the range  $[0, 1]$ .

parameter values,  $\beta_{z^s, x_i}^{s*}$ ,  $\beta_{z^e, x_i}^{e*}$ . Each block of handshake labels is identified by the latent state index in the first column. Only those handshake labels whose normalized parameter value exceeds a threshold (0.01) are retained here for display. The ordering of start and end latent states was chosen so as to display the one-to-one associations that are present between a majority of the start and end latent states inferred by the learning algorithm. These associations arise as a natural consequence of the property that the start and end handshakes are the same in a significant fraction of signs contained in the HSBN training set. The start/end latent state indices for which a one-to-one association were not obtained

are displayed in gray (these are the last 7 start states and the last 5 end states). The probabilities for the start latent state indices given by,  $P(Z^s = z_i^s) := \pi_{\tau=180}$ , are displayed above the corresponding latent state indices. The corresponding probabilities for the end latent state indices are given by,  $P(Z^e = z_i^e) := \mathbf{a}_{\tau=180}^T \pi_{\tau=180}$ .

We note here that the images of handshapes are displayed in the above table only for visualization purposes. The HSBN learning algorithm utilizes the start/end handshape labels annotated for signs in the training set and therefore does not directly leverage configuration/appearance information associated with the handshape labels. The latter information is provided indirectly via the informative priors for the emission distributions,  $\beta_{\tau=1}^{\text{prior}}$ , specified in the first epoch (Section 9.2.1 describes their construction). The priors specified in the first epoch are propagated through the subsequent epochs as described in Section 7.3.

The estimated hyper-parameters for start→end latent state transitions,  $\alpha_{\tau=180}^*$ , are now presented to complete the visualization of the parameters learnt for the HSBN. The state transitions matrix is of size,  $|\mathcal{Z}_{\tau=1}^s| = 34 \times |\mathcal{Z}_{\tau=1}^e| = 29$ , and is displayed in, Figures 10·7-10·9. Start latent states correspond to the rows of this matrix and are ordered in the same sequence as used for  $\beta_{\tau=180}^{s*}$  in Figures 10·4, 10·6. Similarly, the end latent states correspond to the columns of  $\alpha_{\tau=180}^*$  and are ordered in the same sequence as used for  $\beta_{\tau=180}^{e*}$  in the above figures. The handshape label with the highest estimated emission probability is included for each latent state index.

From the above learning results we surmise that the collection of start/end latent states inferred by the HSBNStateSpaceEstimation algorithm provides a relatively compact probabilistic representation for the purposes of modeling the statistical patterns of start/end handshape label pairs and their variations attested among monomorphemic lexical signs contained in the HSBN training set.

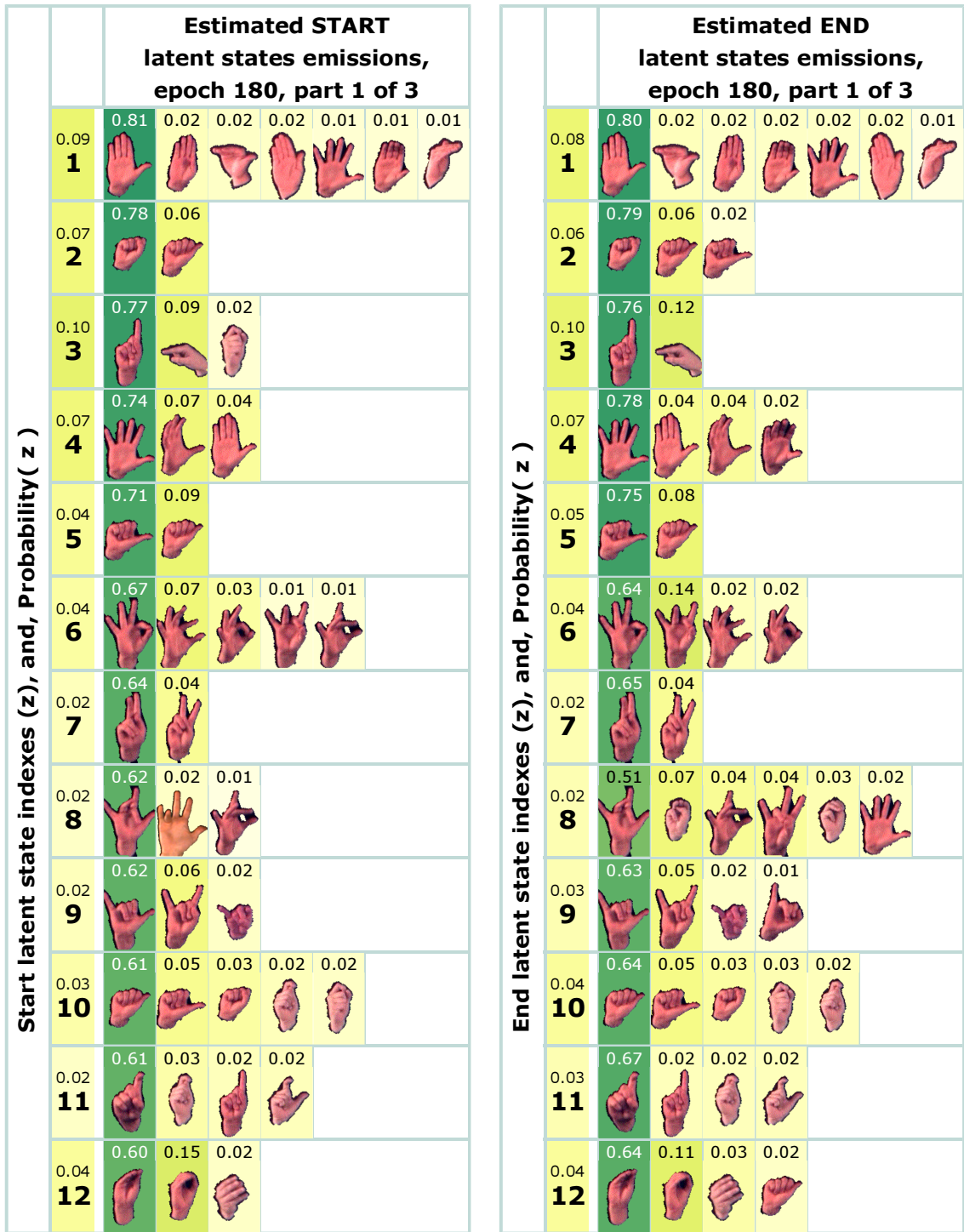


Figure 10-4: Normalized hyper-parameter values,  $\beta_{\tau=180}^{s*}$ ,  $\beta_{\tau=180}^{e*}$ , for emission distributions of start and end latent states estimated in the *final* epoch – part 1 of 3.

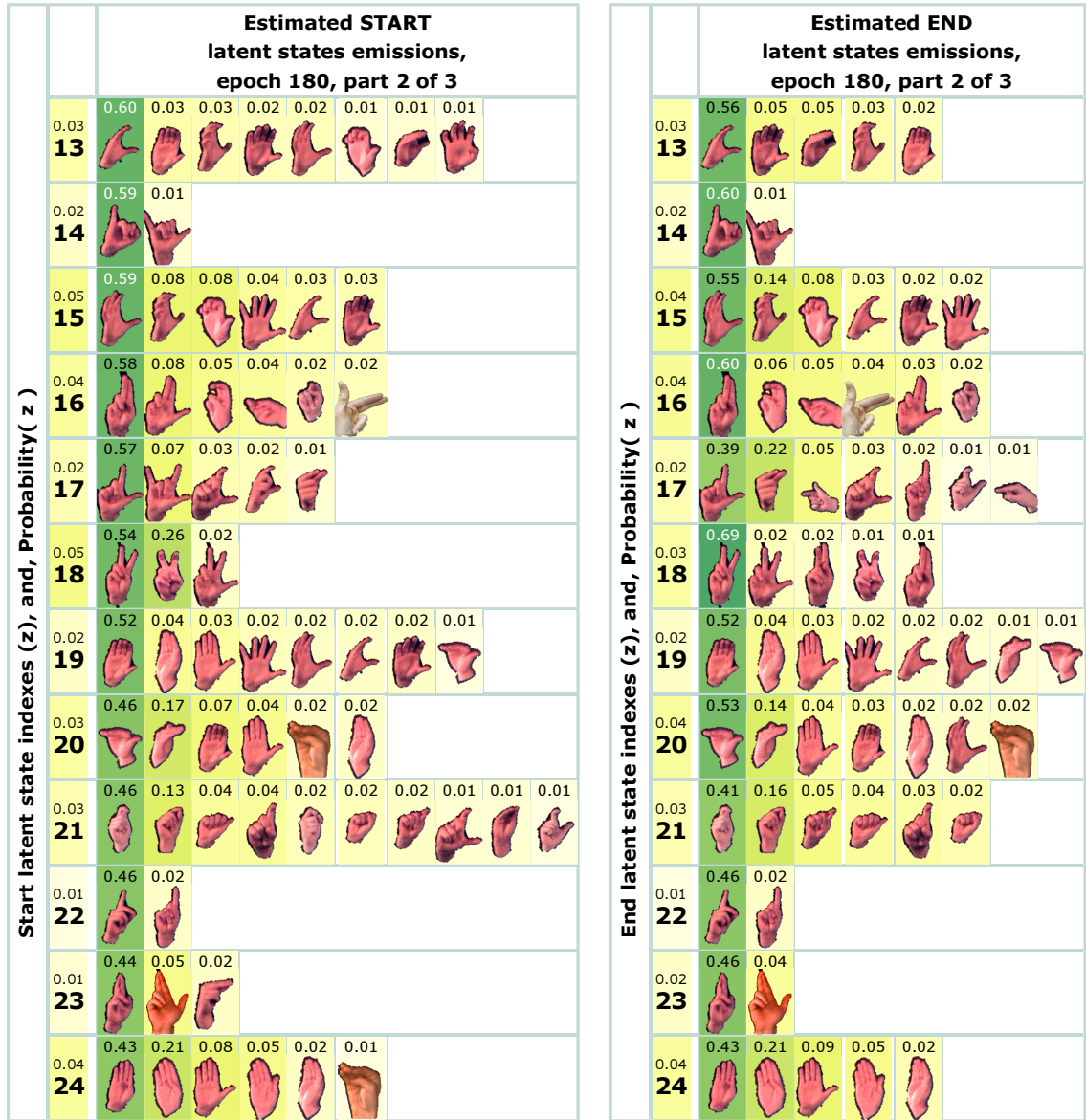
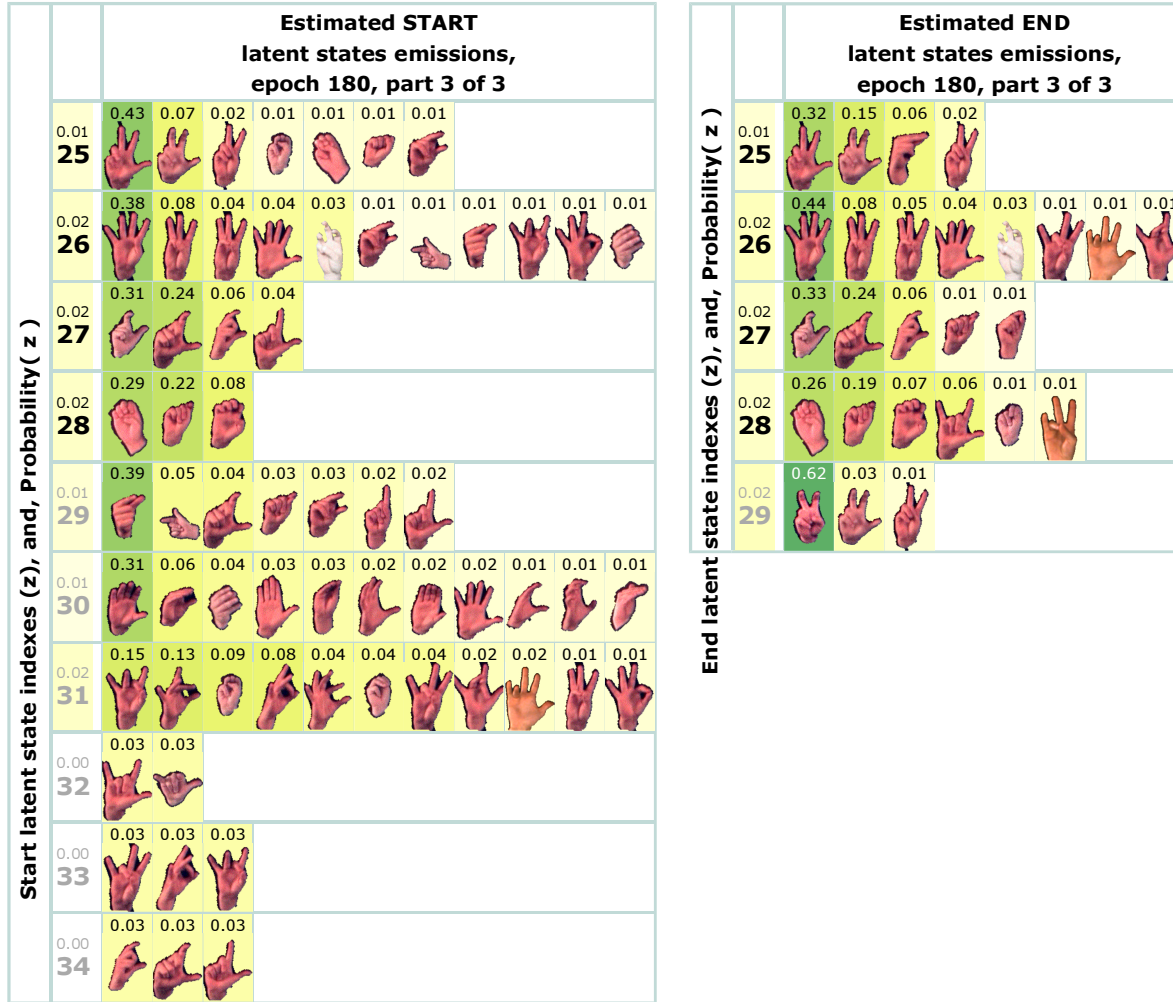


Figure 10-5: Normalized hyper-parameter values,  $\beta_{\tau=180}^{s*}$ ,  $\beta_{\tau=180}^{e*}$ , for emission distributions of start and end latent states estimated in the *final* epoch – part 2 of 3.



**Figure 10-6:** Normalized hyper-parameter values,  $\beta_{\tau=180}^{s*}$ ,  $\beta_{\tau=180}^{e*}$ , for emission distributions of start and end latent states estimated in the *final* epoch – part 3 of 3. The start/end latent state indices for which a one-to-one association were not obtained are displayed in gray.



		Estimated START -> END latent state transition probabilities, epoch 180, part 1 of 3																												
		Estimated END latent states (with top handshape emission for each state)																												
		0.08	0.06	0.10	0.07	0.05	0.04	0.02	0.02	0.03	0.04	0.03	0.04	0.03	0.02	0.04	0.04	0.02	0.03	0.02	0.04	0.03	0.01	0.02	0.04	0.01	0.02	0.02	0.02	0.02
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		
Estimated START latent states (with top handshape emission for each state)	0.09		0.74		0.04			0.01	0.02		0.02	0.01						0.02	0.06											
	0.07		0.01	0.53	0.06	0.13		0.02					0.02		0.03	0.02		0.03		0.01						0.02	0.03			
	0.10		0.01	0.01	0.80	0.01					0.02	0.07										0.01								
	0.07		0.02	0.12		0.57	0.02	0.02			0.04		0.05			0.06				0.01	0.01									
	0.04		0.01	0.01	0.01	0.01	0.76	0.01	0.01	0.01	0.02	0.03	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.04		0.03	0.01	0.01	0.02	0.01	0.73	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.02		0.01	0.01	0.01	0.01	0.01	0.02	0.64	0.01	0.01	0.01	0.01	0.01	0.01	0.05	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.02		0.04	0.01	0.01	0.03	0.01	0.04	0.01	0.60	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.02		0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.68	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.03		0.04	0.01	0.01	0.12	0.04	0.01	0.01	0.01	0.01	0.55	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01
	0.02		0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.65	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.04		0.03	0.01	0.01	0.20	0.01	0.02	0.01	0.01	0.01	0.04	0.01	0.50	0.02	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01

Figure 10-7: Normalized hyper-parameter values,  $\alpha_{\tau=180}^*$ , for start  $\rightarrow$  end latent state transitions estimated in the *final* epoch – part 1 of 3.

		Estimated START -> END latent state transition probabilities, epoch 180, part 2 of 3																													
		Estimated END latent states (with top handshape emission for each state)																													
		0.08	0.06	0.10	0.07	0.05	0.04	0.02	0.02	0.03	0.04	0.03	0.04	0.03	0.02	0.04	0.04	0.02	0.03	0.02	0.04	0.03	0.01	0.02	0.04	0.01	0.02	0.02	0.02	0.02	
		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	
Estimated START latent states (with top handshape emission for each state)	0.03		0.04	0.07	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.61	0.01	0.03	0.01	0.01	0.01	0.01	0.01	0.03	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.02		0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.61	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.05		0.01	0.10	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.03	0.01	0.04	0.02	0.01	0.62	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.04		0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.75	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01
	0.02		0.01	0.03	0.03	0.01	0.04	0.01	0.01	0.01	0.05	0.03	0.01	0.01	0.01	0.01	0.01	0.01	0.45	0.01	0.01	0.01	0.10	0.01	0.01	0.01	0.01	0.01	0.06	0.04	0.01
	0.05		0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.08	0.01	0.45	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.29
	0.02		0.04	0.04	0.01	0.02	0.05	0.01	0.01	0.01	0.01	0.05	0.01	0.05	0.01	0.01	0.01	0.01	0.01	0.01	0.47	0.04	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.03		0.06	0.01	0.01	0.01	0.03	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.05	0.64	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01
	0.03		0.01	0.01	0.03	0.01	0.07	0.01	0.05	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.05	0.01	0.01	0.01	0.57	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	0.01		0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.05	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.06	0.42	0.02	0.02	0.02	0.02	0.02	0.02	0.02
	0.01		0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.58	0.01	0.02	0.01	0.01	0.01
	0.04		0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.03	0.74	0.01	0.01	0.01	0.01	0.01

Figure 10-8: Normalized hyper-parameter values,  $\alpha_{\tau=180}^*$ , for start  $\rightarrow$  end latent state transitions estimated in the *final* epoch – part 2 of 3.

		Estimated START -> END latent state transition probabilities, epoch 180, part 3 of 3																												
		Estimated END latent states (with top handshape emission for each state)																												
		0.08	0.06	0.10	0.07	0.05	0.04	0.02	0.02	0.03	0.04	0.03	0.04	0.03	0.02	0.04	0.04	0.02	0.03	0.02	0.04	0.03	0.01	0.02	0.04	0.01	0.02	0.02	0.02	0.02
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		
0.80	0.79	0.76	0.78	0.75	0.64	0.65	0.51	0.63	0.64	0.67	0.64	0.56	0.60	0.55	0.60	0.39	0.69	0.52	0.53	0.41	0.46	0.46	0.43	0.32	0.44	0.33	0.26	0.62		
	0.43	0.01	0.01	0.01	0.01	0.01	0.05	0.01	0.01	0.05	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.04	0.01	0.01	0.01	0.01	0.43	0.01	0.01	0.04	0.07	
	0.38	0.01	0.03	0.01	0.01	0.01	0.04	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.59	0.01	0.01	0.01	
	0.31	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.06	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.06	0.01	0.01	0.01	0.01	0.01	0.56	0.01	0.01	
	0.29	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.07	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.57	0.01	
	0.39	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.44	0.02	0.02	0.02	0.10	0.02	0.02	0.02	0.02	0.02	0.03	0.02	0.02		
	0.31	0.02	0.05	0.02	0.02	0.02	0.02	0.02	0.02	0.07	0.02	0.41	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	
	0.15	0.01	0.01	0.01	0.07	0.01	0.27	0.01	0.28	0.01	0.04	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.05	0.01	0.01	0.01	
	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	
	0.03	0.03	0.03	0.09	0.03	0.03	0.03	0.09	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	
	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	

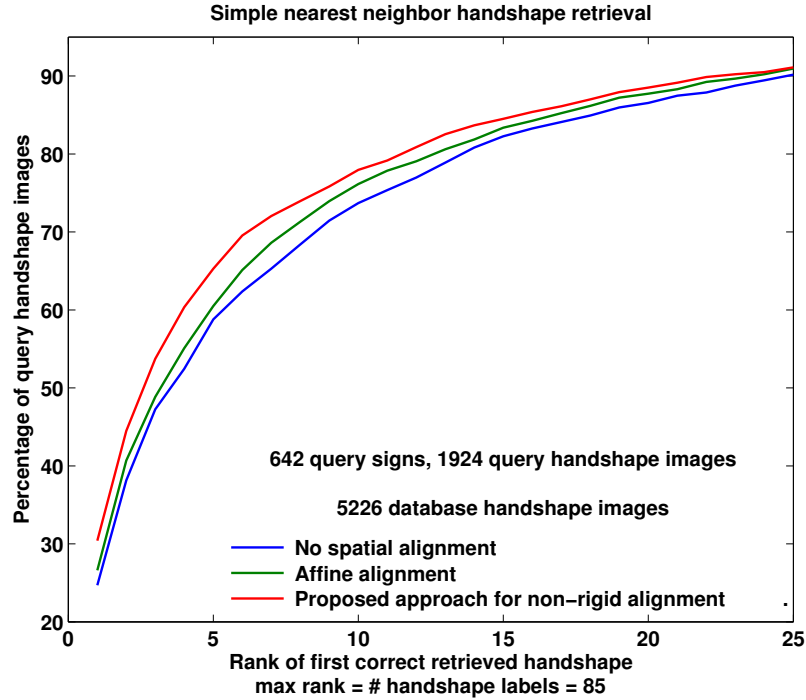
**Figure 10-9:** Normalized hyper-parameter values,  $\alpha_{\tau=180}^*$ , for start  $\rightarrow$  end latent state transitions estimated in the *final* epoch – part 3 of 3. The start/end latent state indices for which a one-to-one association were not obtained are displayed in gray.

## 10.2 Handshape retrieval using image alignment

The implementation of the algorithm for performing nearest neighbor handshape retrieval is described in Section 9.3. Results for handshape retrieval utilizing the ‘no image alignment’, ‘affine image alignment’ and ‘non-rigid image alignment’ methods in computing the similarity scores for handshape image pairs are compared here. The percentage of query handshape images for which the groundtruth handshape label lies within a specified rank among the list of retrieved handshapes is plotted in Figure 10.10. The first occurrence of each handshape label in the retrieved list of handshapes is retained to produce a ranked ordering of handshapes (following the simple-NN ranking procedure described in Section 9.5) for the above plots. The maximum value for the retrieval rank shown on the x-axis is therefore:  $|\mathcal{X}| = 85$ . The top-left corner of this plot corresponds to the performance point for ideal recognition, wherein the handshape labels retrieved at rank 1 in each of the queries corresponds to the groundtruth. The handshape retrieval performance obtained using each of the three methods for computing the similarity scores is tabulated in Table 10.1. From the results shown here, we assess that performing image alignment aids in improving the handshape retrieval performance. The proposed approach for performing non-rigid alignment further improves the ranked retrieval performance compared to an affine alignment method for hand images contained in the HSBN test set. However, further evaluation with different test signers and a retrieval set that spans a larger fraction of signs in the training set is necessary to assess the generalization performance of the three approaches chosen for handshape retrieval.

## 10.3 Handshape inference using the HSBN

We assess the HSBN’s performance for the handshape inference task. The algorithm to perform handshape inference using the HSBN formulation is described in Section 9.4. The impact of several different aspects that have a bearing on handshape inference performance are evaluated in the following experiments. The HSBN parameters estimated in the final learning epoch,  $\tau = 180$ , are used in the first three experiments. The first experiment



**Figure 10-10:** Results of simple nearest neighbor handshape retrieval using different image alignment methods to compute the similarity scores. The fraction of queries for which the retrieved location of the ground-truth handshape label lies within a given rank is shown. Only the first occurrence of each handshape label is retained in constructing the above plot. The maximum value of the rank displayed on the x-axis is therefore  $|\mathcal{X}| = 85$ .

Rank of first correct retrieved handshape (max rank = # handshape labels = 85) → % of queries ↓ (1924 query handshape images)	1	6	11	16	21
No spatial alignment (0.00s avg.)	24.7	62.4	75.4	83.3	87.5
Affine alignment (0.66s avg.)	26.6	65.1	77.9	84.3	88.3
Proposed non-rigid alignment (2.04s avg.)	<b>30.4</b>	<b>69.5</b>	<b>79.2</b>	<b>85.4</b>	<b>89.1</b>

**Table 10.1:** Nearest neighbor handshape retrieval results illustrated in the top plot are summarized in the above table. The highest recognition scores are highlighted in red.

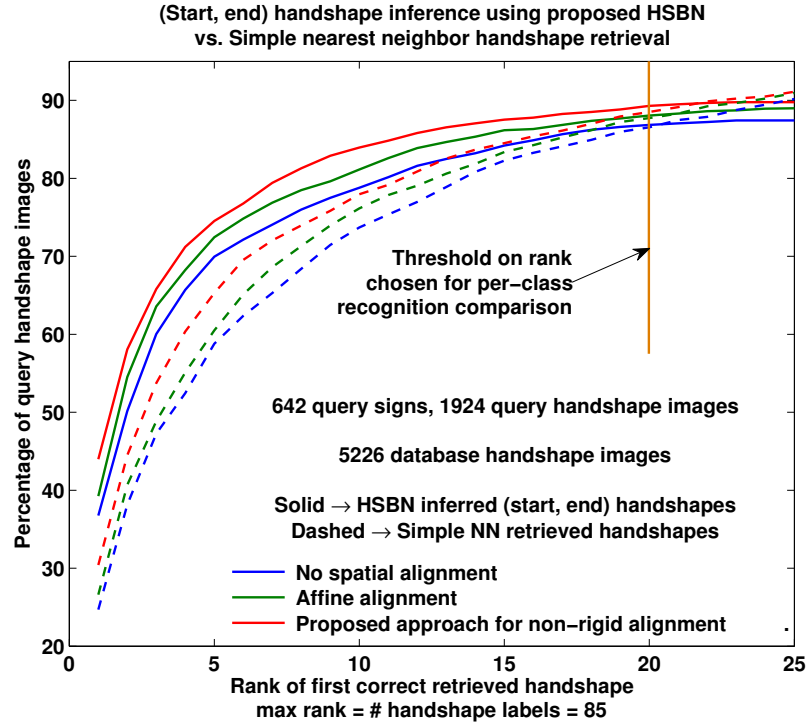
aggregates handshape retrieval/inference accuracies for the entire test set in order to assess the performance of different similarity score computation methods for simple-NN retrieval. The second experiment compares the retrieval/inference accuracies for each handshape class. The third experiment evaluates the retrieval/inference accuracies for **one-handed** and **two-handed:same** handshapes signs in order to compare the performance of the HSBN<sup>dominant</sup> and the HSBN<sup>congruent</sup> formulations. The final experiment evaluates the handshape inference performance using the model parameters obtained through the sequence of learning epochs  $(\omega_{\tau=1}^*, \dots, \omega_{180}^*)$  as produced by the HSBN state-space refinement algorithm.

### 10.3.1 Performance summarized for all handshape classes

In this experiment we evaluate the recognition performance for all signs in the test set. The fraction of start/end query handshapes among signs contained in the test set for which the HSBN inferred rank of the ground-truth handshape label is within a specified rank is displayed using solid lines in Figure 10.11. The results obtained using the simple nearest neighbor method are displayed using dashed lines. The maximum value for the handshape retrieval/inference ranks displayed on the x-axis is,  $|\mathcal{X}| = 85$ . The different similarity scores employed during retrieval are displayed in different colors.

Table 10.2 summarizes the recognition performance for a selected subset of the retrieval/inference ranks. Each column displays the percentage of query handshapes for which the inferred handshape rank lies within a specified value. The first two columns correspond to the retrieval/inference ranks 1, 6. The simple-NN retrieved accuracies are displayed in parentheses for comparison with the corresponding handshape inference accuracies that are shown without parentheses. The best recognition performance in each column is highlighted in red.

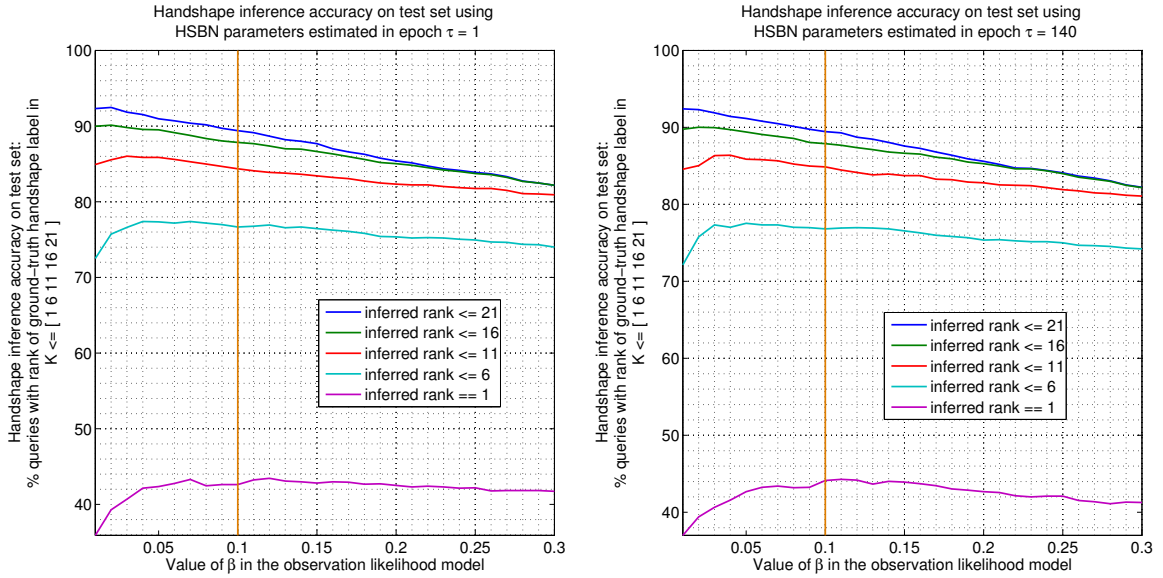
From the above results we assess that the HSBN’s ability to exploit the statistical properties of different handshape combinations and their variations observed in monomorphemic lexical signs aids in the handshape recognition task. However, we also note that handshape inference recognition accuracy using the HSBN is hampered in handshape classes where we



**Figure 10.11:** Performance of HSBN handshape inference summarized over all handshape classes: the ranks for ground-truth handshape labels inferred using the HSBN (solid curves) in query signs are compared to simple nearest neighbor retrieval results (dashed curves) for each of the three similarity score computation methods.

Rank of first correct retrieved handshape (max rank = # handshape labels = 85) → % of queries ↓ (1924 query handshapes)	1	6	11	16	21
No spatial alignment (0.00s avg.)	36.7 (24.7)	72.1 (62.4)	80.1 (75.4)	84.9 (83.3)	87.1 (87.5)
Affine alignment (0.66s avg.)	39.2 (26.6)	74.8 (65.1)	82.6 (77.9)	86.3 (84.3)	88.3 (88.3)
Proposed non-rigid alignment (2.04s avg.)	<b>44.0</b> (30.4)	<b>76.8</b> (69.5)	<b>84.8</b> (79.2)	<b>87.8</b> (85.4)	<b>89.5</b> (89.1)
Rows (with, without) parenthesis := (simple NN retrieval, handshape inference using the HSBN).					

**Table 10.2:** Simple-NN retrieval and HSBN handshape inference results for the entire test set shown in the top plot are summarized in the above table. The highest recognition scores are highlighted in red.

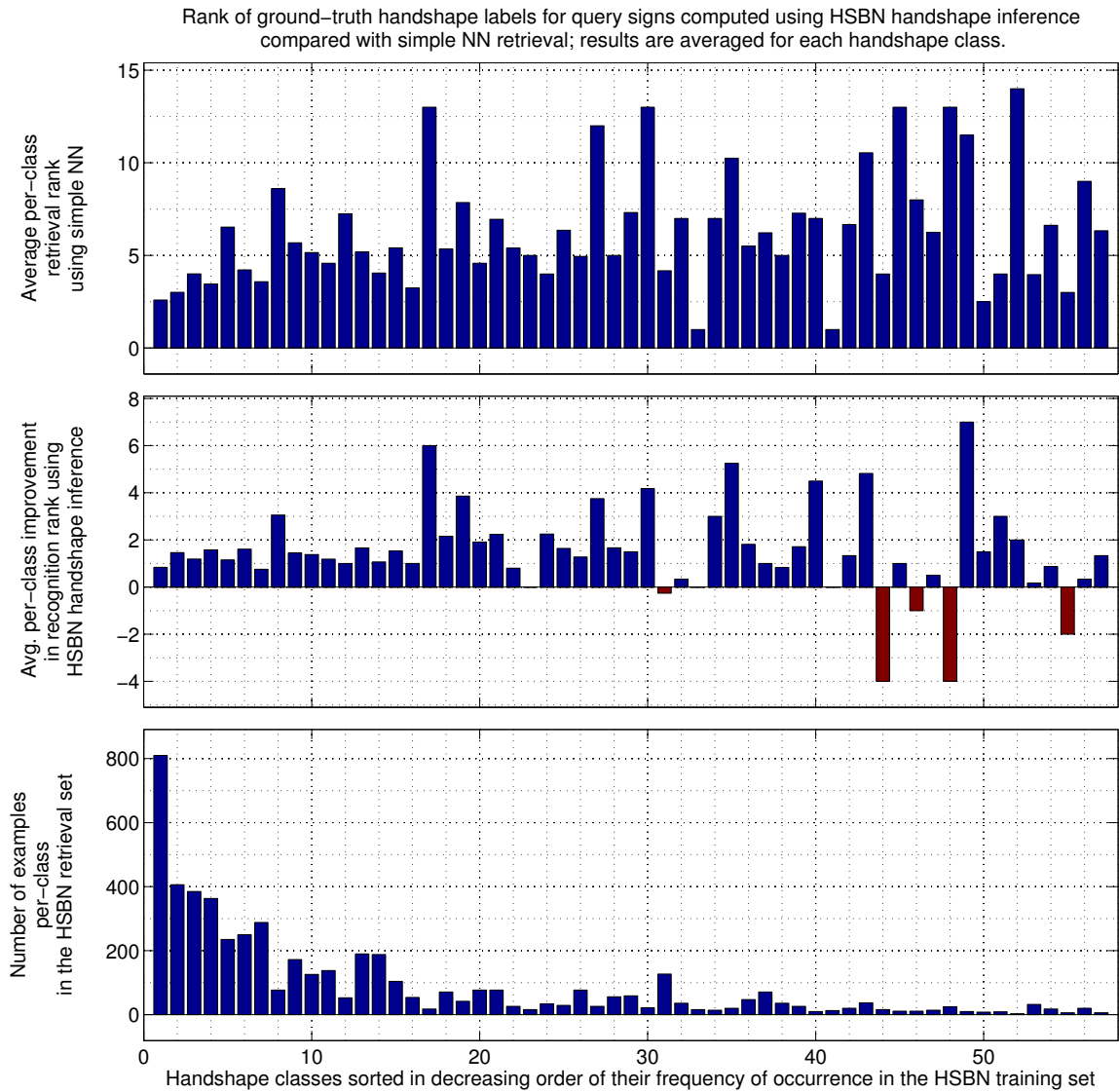


**Figure 10-12:** Impact of the value of  $\beta$  in the observation likelihood model on handshape inference accuracy. Handshape inference accuracies for the test set using HSNB parameters estimated in the first epoch (left chart) and final epoch (right chart) are displayed. The value of  $\beta$  selected for the handshape inference experiments is also displayed.

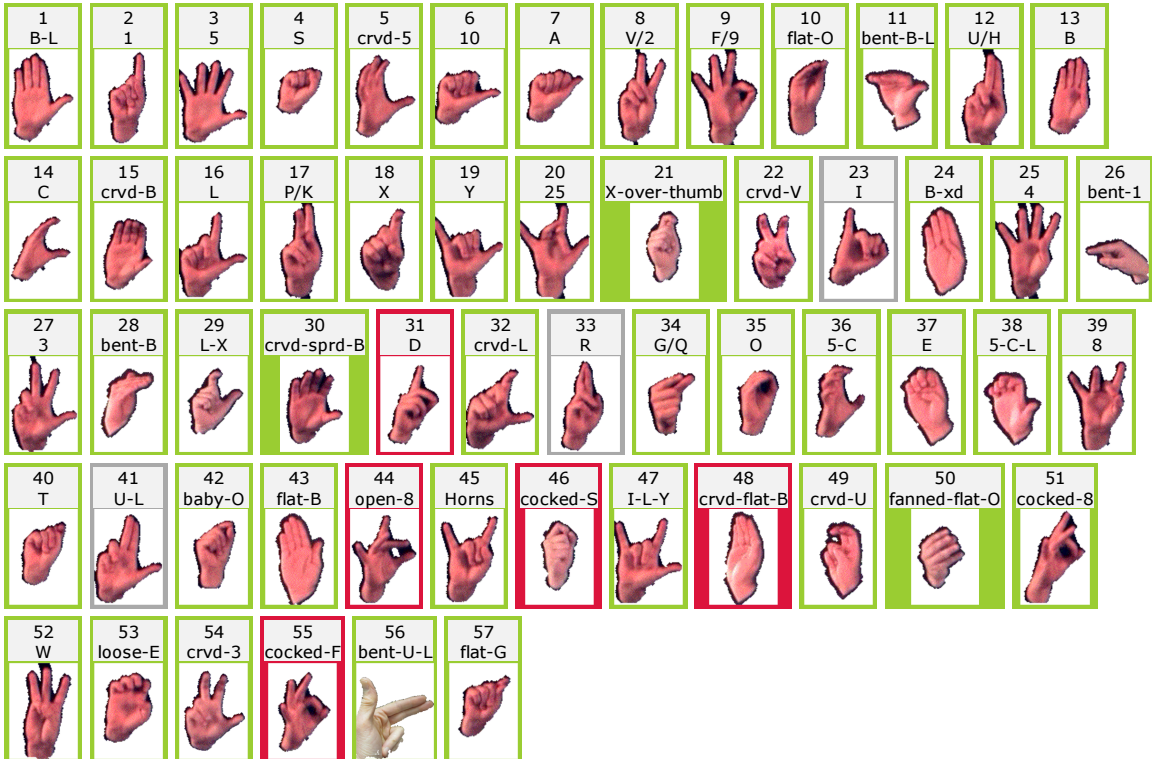
have a relatively small number of examples in the retrieval set. This affects the results obtained for values of retrieval/inference ranks greater than 20 and is investigated in more detail in the next section.

The value of  $\beta$  in the observation likelihood model (Equation 8.1) impacts handshape inference accuracy. The handshape inference accuracy for a selected subset of inferred ranks are plotted against different values of  $\beta$  in Figure 10-12. The left and right charts display the results obtained using HSNB parameters estimated in the first ( $\tau = 1$ ) and final ( $\tau = 180$ ) epochs. The value selected,  $\beta = 0.1$ , for the handshape inference experiments in our current implementation is highlighted. This value was selected based on the handshape inference performance for the final epoch on the test set. In future work a validation set would be used to determine this value. The same value for the size of retrieved sets  $K = 200$  was specified in the simple-NN and HSNB based handshape inference methods.





**Figure 10-13:** Evaluating handshape inference performance for each of the different handshape classes contained in the HSBN test set. In the above charts, the handshape classes on the x-axis are sorted in decreasing order of their frequency of occurrence in the HSBN training set. (a) The top chart displays the average nearest neighbor retrieved rank of the ground-truth handshape labels for the start/end images in the query sign. (b) The second chart displays the improvement (in some handshape classes, an increase) in the average recognition rank after performing HSBN based handshape inference. (c) The third chart displays the number of handshape images for each of the different handshape classes in the HSBN retrieval set. The proposed non-rigid image alignment method was used for handshape retrieval during handshape inference and also for the simple-NN method chosen here for comparison.



**Figure 10-14:** A listing of the handshape classes whose indices appear on the x-axis in the charts displayed in Figure 10-13. Handshapes outlined in green / gray / red correspond to the classes for which HSBN inference demonstrates improvement / retains the same / worsens the handshape recognition rank in relation to simple-NN retrieval.

### 10.3.2 Performance analyzed for each handshape class

In this experiment we evaluate the handshape inference accuracies obtained using the HSBN formulation in each of the different handshape classes. We employ the similarity score computed using the proposed non-rigid image alignment for handshape retrieval and the HSBN model parameters estimated in the final learning epoch for handshape inference. The per-class recognition results obtained are summarized in Figure 10-13. The x-axis in these charts displays handshape classes arranged in the decreasing order of their frequency of occurrence in the HSBN training set (only those handshape classes that occur in the HSBN test set are retained in these charts). The first chart displays the simple-NN retrieved ranks averaged for query images in each of the handshape classes. The second chart displays the

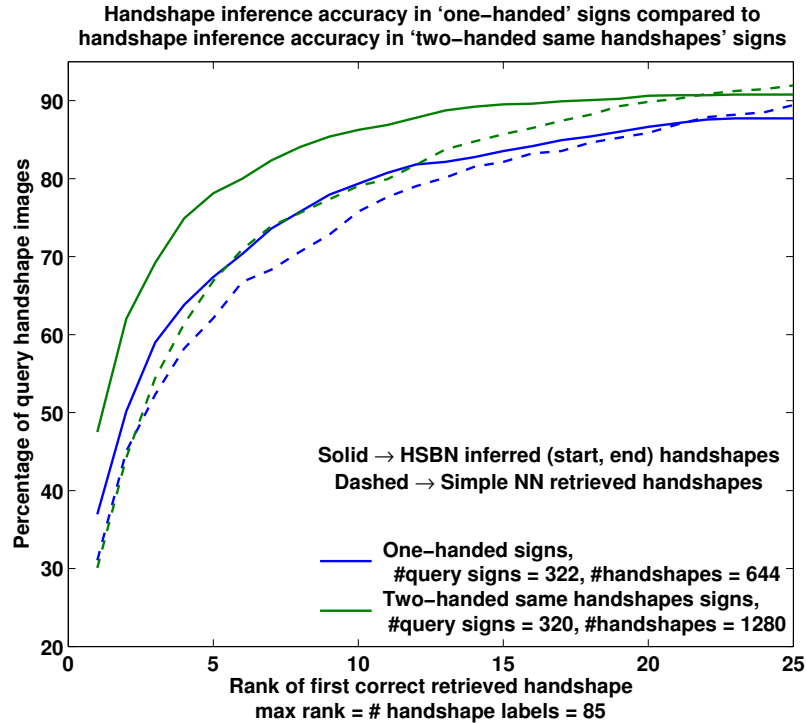
average difference in the recognition ranks between the simple-NN and the HSBN based handshape inference approaches computed using the following expression,

$$\text{average}_{x_q=x}(\text{simpleNN\_rank}(\mathbf{i}_q, x_q) - \text{HSBN\_inferred\_rank}(\mathbf{i}_q, x_q)), \text{ for each } x \in \mathcal{X}^{\text{test set}}. \quad (10.1)$$

Simple-NN supersedes HSBN based handshape inference for large values of retrieval/inference ranks, for reasons that were discussed in the previous section and also observed in Figure 10-11. A threshold of 20 was therefore applied to the retrieval/inference ranks in the above equation in order that certain useful insights can be gained in comparing the two approaches.

The third chart in Figure 10-13 displays the number of examples for each of the handshape classes contained in the HSBN retrieval set.

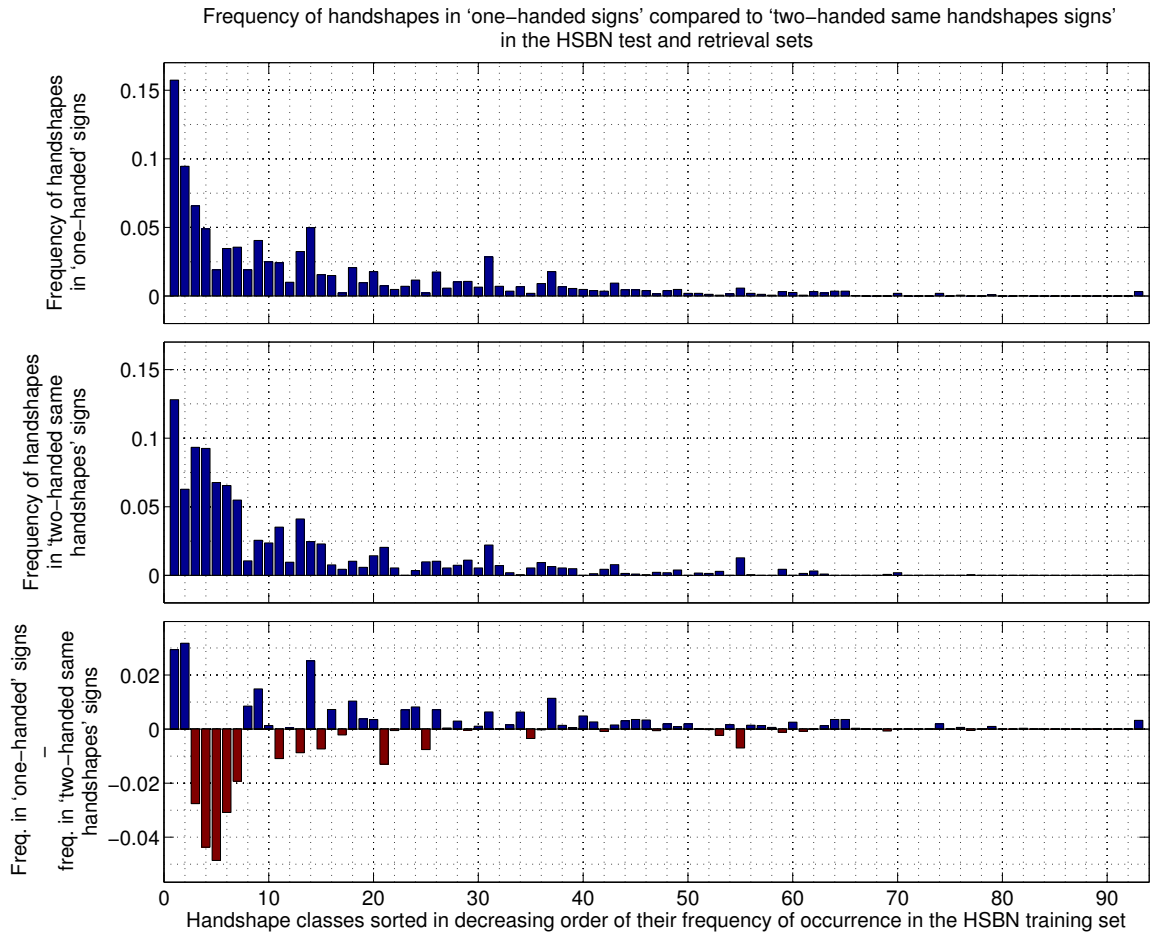
From the above charts we surmise that the handshape classes for which HSBN based handshape inference yields lower ranking results than simple-NN are (in most cases) classes for which the retrieval set contains a small number of examples, e.g., the handshape indices 44, 46, 49 and 57. The handshape classes whose indices are depicted on the x-axis in these charts are enumerated in Figure 10-14. Handshapes outlined in green / gray / red correspond to the classes for which HSBN inference demonstrates improvement / retains the same / worsens the handshape recognition rank. These results are as expected because in handshape classes with a small number of examples in the retrieval set, the observation likelihood formulated in Equation 8.1 is unable to accrue the necessary statistical evidence to boost the nearest neighbor retrieved rank for the ground-truth handshape class label. One possible (time consuming but straightforward) means of addressing these two aspects would be to bring the statistics of the retrieval set in concordance with that of the training set by enlarging the number of signs where we have bounding box annotations for start/end handshapes.



**Figure 10.15:** Handshape inference performance for one-handed query signs are compared to the handshape inference performance for two-handed : same handshapes signs. The corresponding simple nearest neighbor retrieval results are displayed using dashed lines.

Rank of first correct retrieved handshape (max rank = # handshape labels = 85) →					
% of query handshape images ↓	1	6	11	16	21
One-handed signs, #query signs = 322, #handshapes = 644	37.0 (31.1)	70.3 (66.8)	80.7 (77.6)	84.2 (83.2)	87.1 (87.0)
Two-handed same handshapes signs, #query signs = 320, #handshapes = 1280	<b>47.5</b> (30.1)	<b>80.0</b> (70.9)	<b>86.9</b> (79.9)	<b>89.6</b> (86.5)	<b>90.7</b> (90.2)
Rows (with, without) parenthesis :=	(simple NN retrieval, handshape inference using the HSNB).				

**Table 10.3:** Handshape retrieval/inference results for one-handed signs and two-handed : same handshapes signs displayed in the top plot are summarized in the above table. The highest recognition scores are highlighted in red.



**Figure 10-16:** A comparison of frequencies for different handshapes observed in the one-handed (top) and two-handed:same handshapes (middle) signs that are contained in the HSBN test and retrieval sets. The difference in handshape frequency between these two classes is displayed in the last chart. The handshape indices on the x-axis are sorted in decreasing order of their frequencies in the training set. A trend towards handshape classes that are more frequent is observed in two-handed:same handshapes signs.

### 10.3.3 Performance analyzed for two different articulatory classes

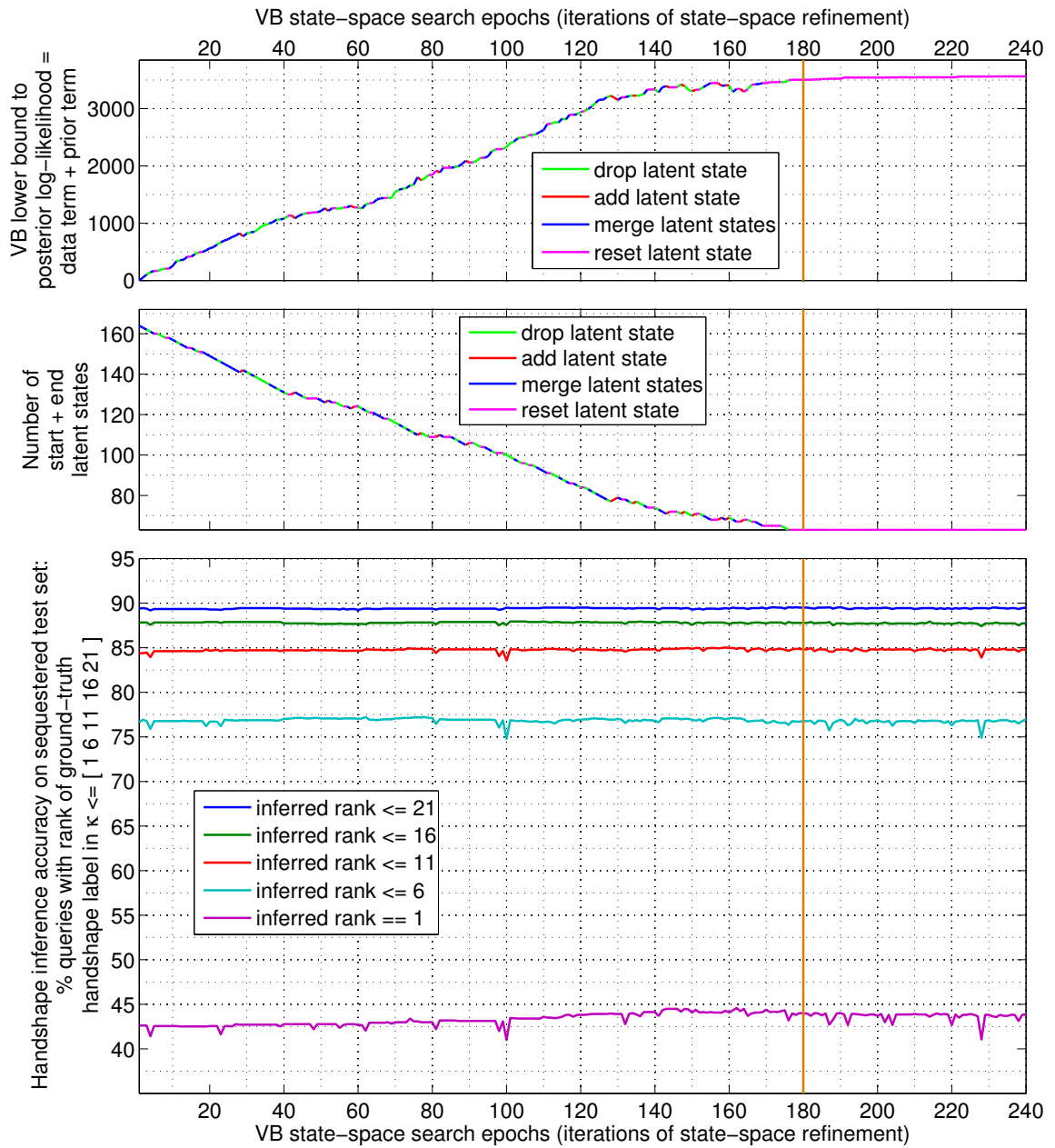
In this experiment we compare the handshape inference accuracies obtained using the two realizations of the HSBN model (the HSBN<sup>dominant</sup> and HSBN<sup>congruent</sup> models) for handshape inference in one-handed and two-handed:same handshapes signs. For this experiment, two-handed:different handshapes in the test set (Table 9.3) are grouped together with one-handed signs (handshapes on the non-dominant hand in these signs are ignored). The simple-NN

retrieval/HSBN inference results for these two classes are plotted in Figure 10.15. The corresponding recognition accuracies are summarized in Table 10.3. Handshape inference using the HSBN improves the recognition performance in both articulatory classes. The improvement observed in `two-handed : same handshapes` signs is markedly higher than in `one-handed` signs. This is as anticipated because the  $\text{HSBN}^{\text{congruent}}$  model for `two-handed : same handshapes` leverages bilateral symmetry constraints for handshapes articulated on the two hands.

We furthermore observe an improvement in simple-NN performance for `two-handed : same handshapes` signs when compared to `one-handed` signs. The underlying reason for this result is the property that handshapes attested in `one-handed` signs in our dataset tend to arise more often from among shapes that occur with less frequency than those in `two-handed : same handshapes` signs. The retrieval set contains a relatively small number of examples for these classes thereby hampering the corresponding simple-NN retrieval accuracy. The frequencies of handshapes attested in the two articulatory classes from among signs in the HSBN test and retrieval sets are compared in Figure 10.16. The first chart displays the frequency of different handshape classes in `one-handed` signs. The second chart displays the corresponding frequencies in `two-handed : same handshapes` signs. The third chart displays the difference in frequencies between `one-handed` and `two-handed : same handshapes` signs. A trend towards more frequent handshape classes (i.e., handshape indices with smaller values) is observed in `two-handed : same handshapes` signs.

#### 10.3.4 Performance analyzed through the learning epochs

In this experiment we evaluate the handshape inference accuracy as a function of the learning epochs employed for state-space refinement in the `HSBNStateSpaceEstimation` algorithm. The handshape inference procedure described in Section 10.3.1 is employed here with the HSBN parameters estimated through the sequence of learning epochs. The handshape inference results for a selected subset of inferred ranks (the same as those selected in the previous experiments) are plotted against the learning epochs in the second chart displayed in Fig-



**Figure 10-17:** Evaluation of the test set handshape inference accuracy as a function of the learning epochs employed for state-space refinement in the HSBNStateSpaceEstimation algorithm. The top and center plots display the evolution of the estimated VBEM lower bound and the total number of latent states through the learning epochs. The last plot displays the fraction of query handshapes from the sequestered test set for which the HSBN inferred rank of the ground-truth handshape label lies within a given value (the five selected values for the inferred ranks are shown in different colors).

ure 10-17. The top and bottom charts display the evolution of the VB lower bound and the estimated number of start+end latent states as were described previously in Section 10.1.

From the plots shown in Figure 10-17 we surmise that the proposed algorithm for HSBN state-space estimation is able to infer a model that uses substantially fewer latent states than the initial model ( $|\mathcal{Z}_{\tau=180}^s| + |\mathcal{Z}_{\tau=180}^e| = 63$  vis-a-vis  $|\mathcal{Z}_{\tau=1}^s| + |\mathcal{Z}_{\tau=1}^e| = 164$ ) without adversely affecting the handshape inference accuracy on the sequestered test set. (The small improvement in rank-1 recognition accuracy is due to the parameter  $\beta$  in the observation likelihood having been tuned for performance in the final epoch; the impact of  $\beta$  on handshape inference accuracy is analyzed in more detail in Section 10.3.1.) Ideally, we would hope to see a significant improvement in the sequestered test recognition accuracy using the more compact representation thereby demonstrating the potential for improvement in the generalization performance as a result of state-space learning. Further efforts are needed however for progress towards this goal. The test and retrieval sets used in our experiments are only a small fraction of the total number of signs available. Expanding these sets to cover a significant fraction of the dataset could yield useful insights with regard to the generalization performance of the proposed learning formulation. Another fruitful direction for investigation would be to evaluate the handshape inference accuracy with different test-users.

### 10.3.5 Examples illustrating HSBN handshape inference results

Start/end handshape inference results produced using the HSBN model for examples of one-handed signs selected from the test set are illustrated in Figure 10-18. We recall that given hand images  $(\mathbf{i}_q^{s:D}, \mathbf{i}_q^{e:D})$  in a one-handed query sign (whose respective ground-truth handshape labels are  $(x_q^{s:D}, x_q^{e:D})$ ), the handshape inference algorithm (Section 9.4) yields a list of start/end handshape label pairs,  $(x_i^{s:D}, x_i^{e:D})$ ,  $1 \leq i \leq |\mathcal{X}|^2$ , arranged in decreasing order of their estimated joint posterior probabilities. For each query sign, the hand images in the first column in the above figure depict the start/end handshapes on the dominant hand in the input video. (The ground-truth start/end handshape labels for the query hand images



are also shown.) The columns 2 – 6 display the top-5 pairs of inferred start/end handshape labels produced using the HSBN. The inferred handshape labels that match the ground-truth are highlighted in green. Handshape instances retrieved using the simple-NN method are displayed for each of the inferred handshape labels. The proposed non-rigid image alignment method for computing hand image similarity scores and the HSBN parameters estimated in the final epoch were used in this experiment.

Handshape inference results produced using the HSBN<sup>congruent</sup> model for examples of two-handed : same handshapes signs selected from the test set are illustrated in Figure 10-18. We recall that given handshape images  $(\mathbf{i}_q^{s:D}, \mathbf{i}_q^{e:D}, \mathbf{i}_q^{s:N}, \mathbf{i}_q^{e:N})$  in a query sign (whose respective ground-truth handshape labels are  $(x_q^{s:D}, x_q^{e:D}, x_q^{s:N}, x_q^{e:N})$ ), the handshape inference algorithm yields a list of start/end handshape label tuples,  $(x_i^{s:D}, x_i^{e:D}, x_i^{s:N}, x_i^{e:N})$ ,  $1 \leq i \leq |\mathcal{X}|^4$ , arranged in decreasing order of their estimated joint posterior probabilities. For each query sign, the hand images in the first column in the above figure depict the start/end handshapes on the dominant and non-dominant hands in the input video. The columns 2 – 6 display the top-5 tuples of inferred handshape labels produced using the HSBN<sup>congruent</sup> model along with the handshape instance retrieved using the simple-NN method.

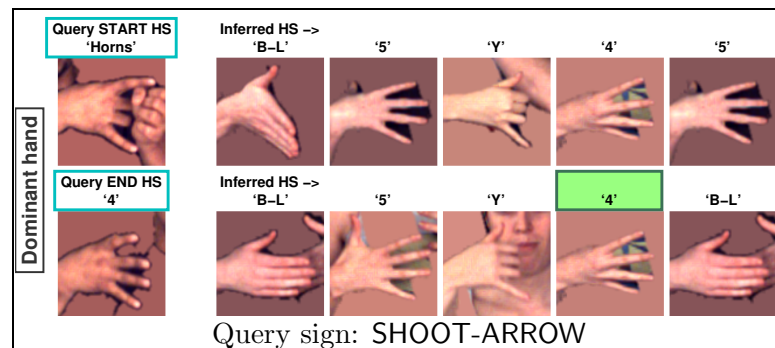
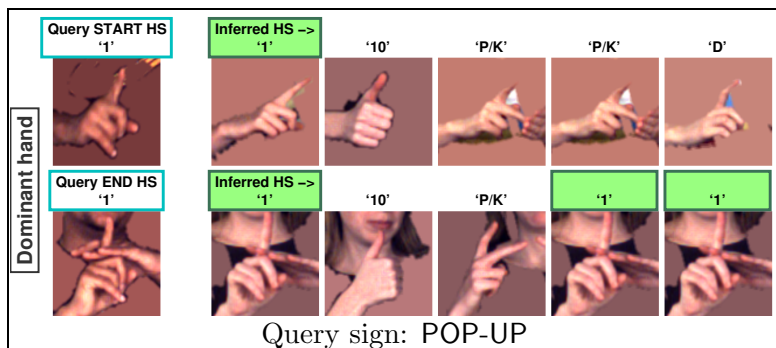
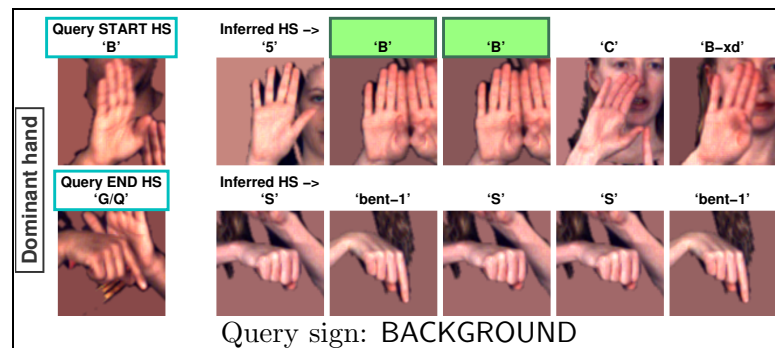
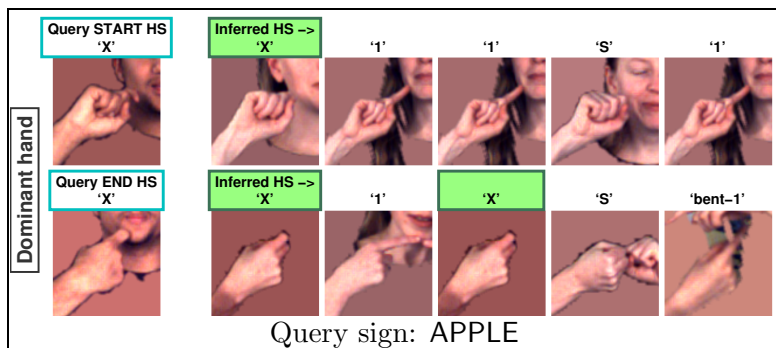
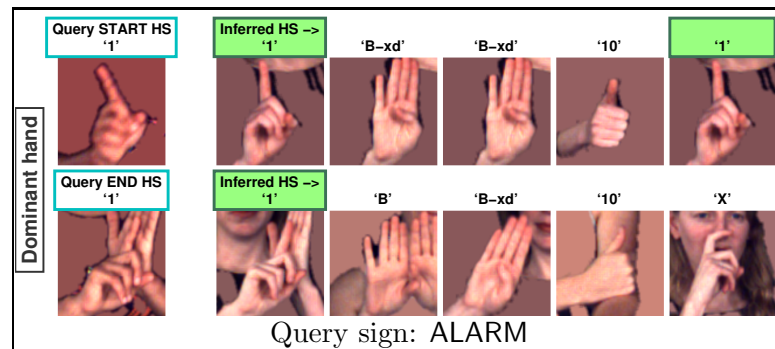
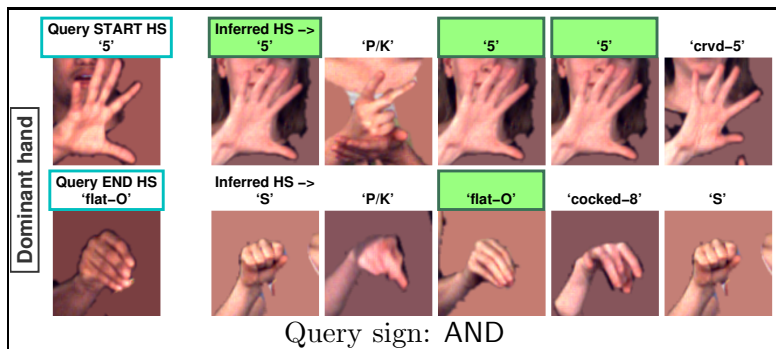


Figure 10.18: Examples of results for start/end handshape inference in one-handed signs using the HSBN.

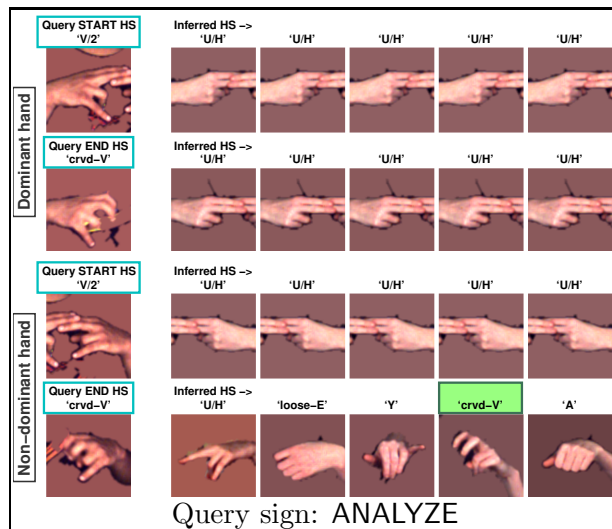
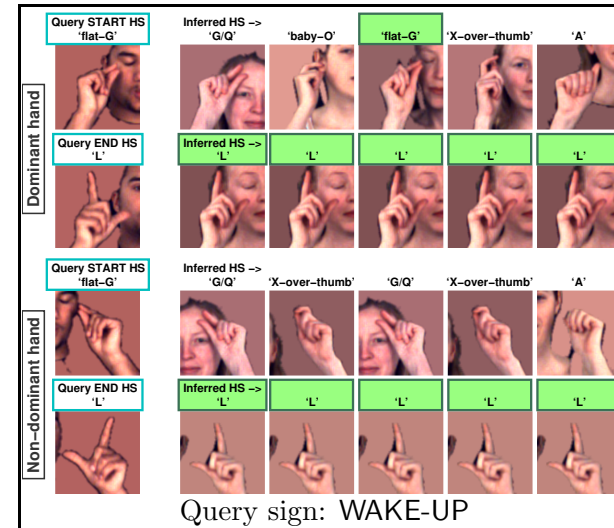
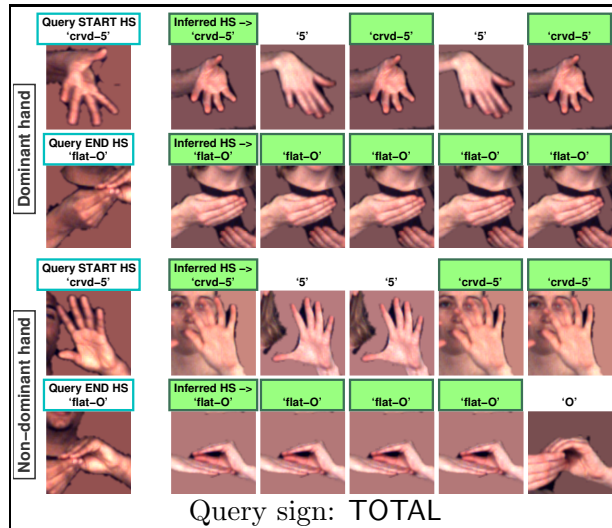


Figure 10-19: Examples of results for start/end handshape inference in two-handed:same handshapes signs using the HSNB<sup>congruent</sup> model.

## 10.4 Discussion

In our empirical evaluation of the HSBN conducted using the ASLLVD dataset towards facilitating progress towards person-independent handshape recognition in sign language video we observe the following consistent trends. Among the different similarity score computation methods that were employed, the proposed non-rigid alignment method demonstrates an improvement for simple nearest neighbor handshape retrieval. When averaged over all handshape classes, HSBN based handshape inference outperforms the simple nearest neighbour method for retrieval/inference ranks  $\leq 20$ .

An improvement in ranked recognition scores was observed in both `one-handed` and `two-handed : same handshapes` signs. The improvement seen in the latter class is noticeably higher because an additional pair of start/end handshape observations from the non-dominant hand are available to use within the `HSBNcongruent` model (the `HSBNcongruent` model exploits the properties of bilateral symmetry in handshapes articulated in these signs). In addition, the start/end handshapes in `two-handed : same handshapes` signs in our dataset tend to arise more often from among the higher frequency handshape classes than in `one-handed` signs, which in-turn boosts the recognition scores for both the retrieval and inference methods.

Comparing the ranked performance of handshape inference with simple-NN for each handshape class also reveals certain trends. In handshape classes where either very few examples are available in the retrieval set or where the statistics for handshapes in the retrieval set are markedly lower than those in the training set, HSBN based handshape inference on average yields lower ranks than simple-NN. Enlarging the retrieval set to encapsulate a larger fraction of signs in the training set provides one potential direction to address this deficit.

To briefly summarize the impact of the different components in the proposed handshape inference algorithm, simple-NN with no image alignment yields a rank-1 retrieval accuracy of 24.9%, simple-NN using the proposed non-rigid alignment method yields a rank-1 retrieval accuracy of 30.5%, while HSBN handshape inference leveraging linguistic constraints yields a rank-1 recognition accuracy of 43.4%. These are nevertheless modest numbers – person-

independent handshape recognition in sign language video remains a challenging problem for computer vision approaches. The resolution of hand images in our dataset are also on the low side ( $90 \times 90$  pixels) when compared to the resolutions that are available from current imaging devices. The bounding box precision has a significant impact on the handshape retrieval performance especially in cases where the hand is articulated close to the face or to the other hand. In future work, we intend to investigate the handshape recognition accuracy using results obtained from an automatic hand location detection and tracking method. One specific approach for hand location detection in sign language video was evaluated in [Thangali and Sclaroff, 2009].

The final evaluation we performed was to assess the impact of the proposed HSBN-StateSpaceEstimation algorithm for learning the hidden variable state-space,  $\hat{\mathbf{Z}} = (\hat{\mathbf{Z}}^s, \hat{\mathbf{Z}}^e)$ , in the HSBN. With careful choices for hyper-parameters of the prior distributions and the hyper-parameters for model initialization in the first epoch, the HSBNStateSpaceEstimation algorithm in the trial shown starts with a set of 164 start+end latent states and through the process of maximizing the variational Bayes lower bound infers a collection of 63 start+end latent states in the final epoch, 180. The epoch to conclude the learning was chosen based on the observed profile for the VB lower bound to avoid over-fitting. The handshape inference performance on the sequestered test set remains constant through the learning epochs thereby providing evidence that the state-space estimation algorithm is able to produce a relatively concise yet accurate (in the specific sense of being able to retain sequestered test accuracy through the learning epochs) representation for modeling the patterns of start/end handshape sequences and their attested variations. Further evidence for potentially beneficial aspects of the state-space learning formulation were obtained by a qualitative analysis of the properties of the emission distributions associated with the inferred latent states in the final epoch and by comparing these properties with the corresponding properties estimated for the initial latent states.

The results presented from experiments conducted so far are limited in terms of generalizability since we have restricted our attention to one specific test-user from among the

six signers who provided signs for the ASLLVD.

## Chapter 11

### Discussion and Future work

In this thesis we focused on the problem of start/end handshape recognition in monomorphemic lexical signs. We formulated the HSBN as a Bayesian network model to represent the properties of start/end handshape sequences and their attested variations in monomorphemic lexical signs. The HSBN is designed to aid in the start/end handshape inference problem wherein given start/end hand images (on either the dominant hand in one-handed signs or on both the dominant and non-dominant hands in two-handed signs) as input, labels from among a predefined set of handshape configurations are desired as output. The set of handshape labels were selected by linguists for the purposes of preparing ASL annotations. A dataset that contains a reasonably large collection of signs from multiple native sign language users annotated with linguistic properties was prepared in order to acquire the required data for the purposes of training and evaluation of the HSBN model. Data elicitation, annotation, and analysis were carried out by Carol Neidle and linguistics students at Boston University.

The HSBN employs hidden variables to encapsulate the properties of sign-independent variation in start/end handshapes among different productions of signs in the vocabulary. Learning the HSBN involves estimating the state-space for the hidden variables (represented here as a collection of discrete states) and the parameters of the probability distributions for the transitions between start and end hidden states along with the respective start and end emission distributions associated with these hidden states. The variational Bayes lower bound to the total training data log-likelihood [Beal, 2003] was employed as the objective to maximize during the learning. Given the learned HSBN model parameters, different realizations of the HSBN are constructed in order to represent the properties of start/end

handshapes on the dominant hand in `one-handed` and on the dominant+non-dominant hands in `two-handed:same handshapes` and `two-handed:different handshapes` articulatory classes. The observation likelihoods required during handshape inference were obtained using a  $k$ -nearest neighbor approach that incorporates non-rigid image alignment between hand images. A retrieval set consisting of start/end handshape images from a subset of signs in the training set was constructed in order to compute the observation likelihoods.

Handshape inference results obtained for a sequestered test are along the lines expected for a statistical learning formulation. Simple nearest neighbor retrieval was chosen as the baseline method for comparison. While HSBN based handshape inference improves the handshape recognition accuracy in a majority of handshape classes, in handshape labels for which the retrieval set contains relatively few examples, HSBN inference yields lower accuracy than simple-NN. Enlarging the retrieval set could help alleviate this deficit. Handshape inference accuracies for `two-handed:same handshapes` signs show a marked improvement over `one-handed` signs. This is as expected because the HSBN<sup>congruent</sup> model leverages the properties of bilateral symmetry in handshape articulation. Furthermore, the statistics of handshapes in signs from this class are skewed towards more frequently occurring handshape configurations which in-turn boosts the performance of both simple-NN retrieval and HSBN inference methods.

The final experiment evaluates the HSBN state-space estimation algorithm. Start/end hidden variable states in the HSBN model are initialized to the set of all handshape labels. An appropriate set of hyper-parameters for the prior distributions were also provided during model initialization. Through the process of maximizing the variational Bayes lower bound, the proposed state-space estimation algorithm is able to infer an optimized representation that employs a substantially smaller number of total latent states than in the initial model (less than half the total number chosen for initialization in the trial shown). Handshape inference performance on the sequestered test set remains unaffected, however. This therefore provides evidence that the state-space learning method is able to retain the properties of start/end handshape sequences and start/end handshape variations as are essential for



handshape inference.

## **11.1 Limitations of the proposed formulation**

We identify some of the different areas where the proposed approach for handshape inference leaves room for further enhancements.

### **11.1.1 Assumptions in the HSBN representation**

Two important assumptions that were made in order to simplify the HSBN model are described below.

#### **Localization of start and end video frames**

In our current formulation, we assume that start/end frames that accurately localize a sign in the video sequence are provided as inputs during handshape inference. Selecting a specific pair of start and end frames for a sign can be challenging, especially in signs where a change in handshape is observed between the start and end points of a given sign. Allowing multiple video frames to be utilized for the start and end points of a sign provides one possible means of addressing this issue.

#### **Missing relationships to include between handshape variables**

The HSBN representation utilizes a pair of start/end latent variables to model the properties and constraints that relate the start/end handshape variables in monomorphemic lexical signs allowing for certain phonological variations in handshape. The tree structured representation assumed for the HSBN does not fully encapsulate many of the linguistic constraints among the handshape variables. Including additional links (i.e., conditional distributions) among handshape variables in the HSBN model suggests one possible means of improving the representational power of the model.

### **11.1.2 Learning the HSBN model**

Some of the limitations with respect to the formulation developed for learning the HSBN are categorized below.

#### **Annotations for handshapes**

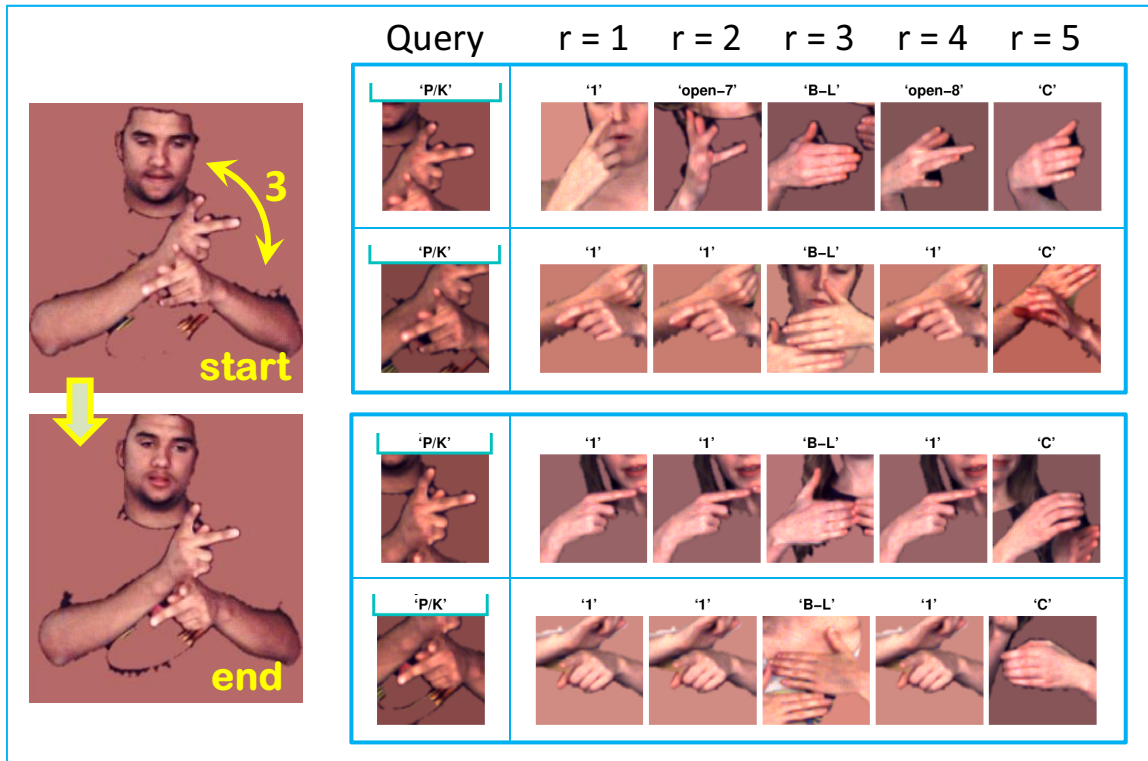
During the HSBN learning, start/end handshape annotations prepared by linguists are assumed as ground-truth for the start/end hand configurations. In many cases, however, there is a degree of uncertainty in the annotated handshapes (for reasons outlined in Section 4.3.1). Utilizing the start/end hand images from video sequences contained in the training set in addition to the handshape annotations for learning the HSBN provides one possible means of accommodating the uncertainty inherent in the handshape annotations.

### **Specification of priors during the learning**

In the learning trials that were conducted, the convergence properties of the learning algorithm were influenced by the scalar concentration parameter for the Dirichlet priors. A cross-validation technique is necessary to determine an appropriate value for the concentration parameter. The concentration parameter was chosen empirically in our experiments based on the properties of the evolution of the data log-likelihood and prior terms comprising the variational Bayes lower bound through the learning epochs. Other families of priors can provide a greater degree of control than the Dirichlet over the model parameters associated with the low frequency handshape classes. An example of one such prior is discussed in the future work section.

### **Optimization objective maximized when learning the HSBN**

The variational Bayes lower bound objective maximized during the HSBN learning was developed within a fully Bayesian (therefore, generative) framework. Linguistic distinctions between different items in the vocabulary conveyed by handshapes are not explicitly leveraged in the proposed HSBN learning algorithm. As a consequence, certain latent states become associated with unexpected handshape productions (these handshapes tend to occur with a relatively low frequency in the training set) in performing the HSBN state-space refinement. A discriminative learning formulation for estimating the properties of latent states while ensuring that the distinctions between certain handshape classes are retained is one possible approach (in addition to a different choice for the prior) towards improving the emission properties of the estimated latent states.



**Figure 11.1:** An example of a query sign where HSBN handshape inference fails to produce acceptable results because nearest neighbor retrieval for each of the query hand images does not succeed in retrieving the correct handshape among the top-200 results.

### 11.1.3 Observation likelihoods for start/end hand images

A handshape retrieval based method was employed in this research as one simple approach for computing the observation likelihood distribution. Some of the problems pertaining to the computation of the observation likelihoods in this fashion are described below. Figure 11.1 illustrates these difficulties using one particular query sign as an example.

#### Start and end hand locations

Hand location bounding box annotations were assumed as inputs for the handshape inference experiments conducted here. The accuracy of the hand location bounding boxes has a measurable impact on the handshape retrieval results. Assessing handshape inference performance with hand locations obtained using an automatic hand location detection and tracking method is one important aspect that we intend to investigate in our future work.

### **Weighting term for combining handshape retrieval results**

The observation likelihood distribution employs an exponential weighting term based on the rank of the retrieved handshape matches (Equation 8.1) in order that different image alignment methods can be compared directly. A weighting term that incorporates the similarity score computed between the query and the retrieved handshape matches can provide one means of improving the observation likelihood.

### **Robustness to clutter observed in hand images**

Significant sources of clutter within the hand image bounding box include the presence of the other hand or the face. Segmenting the hand using motion cues provides one means of addressing the problems that arise as a consequence of clutter within the hand image.

### **Robustness to differences in the properties of the hand among different signers**

Even with a non-rigid image alignment method to accommodate differences in the anthropometric properties of the hand among different signers, person-independent handshape retrieval presents significant challenges. A method for computing the similarity score that does not rely on computing an explicit spatial alignment between a pair of images is one possible alternative approach that can aid with improving the handshape retrieval accuracy.

### **Discriminative features for handshape retrieval**

The proposed approach for handshape retrieval does not utilize features specifically learnt for distinguishing between different handshape classes. A discriminative approach for handshape classification can aid in addressing the problem that handshapes which differ in configuration but share very similar appearance appear frequently in the top-ranked list.

## **11.2 Future work**

Topics for future investigation are organized into those that pertain to enhancements of the proposed HSBN formulation and those that pertain to experiments to further assess the HSBN performance.

### **11.2.1 Enhancements to the formulation**

Possible enhancements to the HSBN formulation are described below.

### **Patterns of variation (free vs. context dependent)**

Several factors contribute to variations observed in signs. The focus of our efforts in this thesis was on modeling the properties of phonological variation produced in handshapes articulated within isolated signs. From a linguistics perspective, variations that arise as a consequence of the phonological environment in which the handshape appears are particularly interesting. In compound signs, for example, variations in start/end handshapes are frequently produced as result of co-articulatory influences from the preceding/succeeding sign segments. Extending the HSBN to model co-articulatory phenomena in compound signs is an intriguing direction that we intend to pursue in future work. The ASLLVD includes a modest number of compound signs along with start/end handshape annotations for morphemes contained in these signs and can therefore provide the data that facilitates in modeling co-articulatory phenomena.

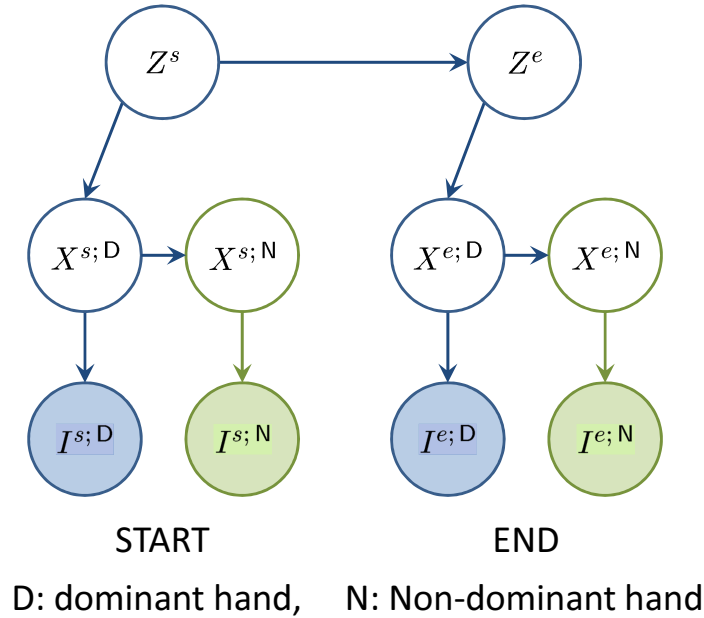
### **Enhancements to the model structure**

Incorporating frames that are adjacent to the start and end frames of a sign within the HSBN representation can provide one possible means of enhancing the robustness of the representation to errors in localizing the sign temporally within a given video sequence. Furthermore, the hands in start/end frames can be more strongly occluded than in the intermediate frames (as is the case for the example shown in Figure 11.1). Observations from adjacent frames can therefore be leveraged towards improving the handshape inference accuracy.

We employed a tree structured representation for the HSBN that was motivated by linguistic considerations. Including a different set of dependencies, for instance, between pairs of start/end handshape variables  $(X^{s:D}, X^{e:D})$ ,  $(X^{s:N}, X^{e:N})$  as illustrated in Figure 11.2, can enable the model to more accurately reflect the linguistic relationships among handshapes articulated on the dominant and non-dominant hands.

### **Priors for the model parameter distributions**

The choice of an appropriate prior has a significant impact on the properties of the HSBN model parameters estimated by the proposed learning algorithm. The current formulation



**Figure 11.2:** An alternate formulation of the HSBN to more directly represent the dependencies between the handshapes articulated on the non-dominant hand and that of the dominant hand in two-handed : same handshapes signs.

relies on Dirichlet priors for the model parameters. Other families of priors, such as the Pitman – Yor process [Teh, 2006] can be a more appropriate choice to accommodate the properties of handshape classes that occur relatively infrequently in the dataset (the author wishes to thank Erik Sudderth for this suggestion).

### Generative vs. discriminative learning formulations

Discriminative learning approaches for structured representations (e.g., Deformable Part Models [Felzenszwalb et al., 2010], Structured Prediction Cascades [Weiss et al., 2010], Conditional Random Fields [Morency et al., 2007]) have been demonstrated to yield substantial gains in performance over purely generative approaches and therefore offer an interesting future direction to leverage discriminative information within the HSBN learning formulation.

### Enhancements to the observation likelihood model

To further improve the nearest neighbor based retrieval approach, a more flexible feature representation, in terms of relaxing the lattice constraint for the feature locations, is needed

for the observation likelihood model to incorporate robustness to bounding box inaccuracies obtained from a hand location detection and tracking method. The need for computing an explicit image alignment may also be reduced when a sufficiently large database of hand images is available, or, in a discriminative approach for handshape classification that utilizes a different feature representation. Incorporating depth input produced by RGB+D cameras is another venue for investigation.

Another area where a discriminative approach can prove beneficial is in formulating the observation likelihood model. An efficient method for handshape classification (such as, for example, the decision forest algorithm [Shotton et al., 2011]) can help circumvent the computational expense required to perform nearest neighbor handshape retrieval incorporating hand image alignment.

### 11.2.2 Empirical assessment

An empirical evaluation employing different signers in the lexicon dataset as test users is needed to more carefully assess the generalization performance of the HSBN formulation. This is relatively straightforward to implement given the hand location bounding boxes for video sequences from different signers.

An implementation of the HSBN for handshapes articulated on the non-dominant hand in two-handed : different handshapes signs is needed to complete the different components of the proposed formulation. Since the non-dominant hand in these signs only takes a small range of possible handshapes a relatively compact model can be learnt.

Using handshapes inferred by the HSBN for performing sign retrieval from the lexicon dataset will give us with a baseline for the extent to which handshape inference alone is beneficial for sign retrieval. Handshape inference using the HSBN produces a ranked list of different handshape tuples with the highest posterior probability, whereas only one specific tuple of handshapes is available for a sign on the database side. Therefore an extension to the handshape inference approach is necessary to accommodate this difference.

The HSBN representation can also be used to infer the articulatory class of an input sign.

Handshape inference likelihoods computed using the two different HSBN representations for the input sign can provide a method for distinguishing two-handed : same handshapes signs from two-handed : different handshapes signs.

We now summarize some of the experiments necessary to evaluate the HSBN performance in more general environments.

Automatic hand location detection and tracking is essential in order to evaluate the feasibility of the proposed handshape inference approach under more realistic conditions. In earlier work [Thangali and Sclaroff, 2009] we described a method for hand location detection in sign language video. This approach, however, was computationally expensive. A decision forest based approach [Shotton et al., 2011] provides one appropriate framework to train computationally efficient hand location detectors.

Given the trajectory of hand locations, sign retrieval from the lexicon dataset can be performed by matching both the inferred handshapes as well as the hand movement trajectory. [Dreuw et al., 2006, Alon et al., 2009] are two among many approaches that have employed dynamic time warping for sign retrieval.

Depth input from RGB+D sensors can facilitate in segmenting hands from the background as well as assist with tracking hand locations in the input signing sequence. Furthermore upper body pose estimation facilitated by using depth inputs and can provide additional features for sign retrieval.

An approach for handshape classification trained in a discriminative fashion can yield benefits both in terms of improved accuracy as well as reduction in the computation required during handshape inference. In order to train a handshape classifier, however, the different 3D hand orientations need to be either explicitly or implicitly accommodated.

Handshape inference in compound signs, and, in continuous signing video sequences presents an interesting avenue to demonstrate handshape inference results. This, however, requires an extension of the HSBN representation to explicitly model the co-articulatory effects.



### 11.3 Summary

In this thesis we aimed to demonstrate the benefits of leveraging linguistic properties of handshape articulation in monomorphemic lexical signs within a probabilistic representation for the handshape inference problem. Estimating the HSBN parameters in a data-driven formulation presents many interesting questions especially those that pertain to modeling the properties of handshape variations that are produced as result of general language processes. Some of these questions were addressed in this research by employing the variational Bayes lower bound as the objective to maximize during model estimation. The HSBN yields a measurable improvement over the baseline simple nearest neighbor method in a person-independent large vocabulary handshape recognition task. We envision that some aspects of the proposed formulation for handshape inference can be leveraged for modeling other articulatory parameters in a sign language recognition system, for example, in modeling the properties of start/end hand locations and hand orientations within monomorphemic lexical signs.

The collection and preparation of the ASL lexicon video dataset played an instrumental role in enabling the implementation of the proposed approaches for learning and evaluation of the HSBN model. We anticipate that because of the extensive linguistic annotations that are available for signs contained in this dataset, the ASLLVD can provide a valuable resource for furthering research into data-driven methods for sign language recognition.

## References

- [Alon et al., 2009] Alon, J., Athitsos, V., Yuan, Q., and Sclaroff, S. (2009). A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1685–1699.
- [Athitsos, 2006] Athitsos, V. (2006). *Learning Embeddings for Indexing, Retrieval, and Classification, with Applications to Object and Shape Recognition in Image Databases*. PhD thesis, Boston University.
- [Athitsos et al., 2008a] Athitsos, V., Alon, J., Sclaroff, S., and Kollios, G. (2008a). Boost-Map: An embedding method for efficient nearest neighbor retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):89–104.
- [Athitsos et al., 2008b] Athitsos, V., Neidle, C., Sclaroff, S., Nash, J., Stefan, A., Yuan, Q., and Thangali, A. (2008b). The American Sign Language lexicon video dataset. In *Proceedings of Workshop on Computer Vision and Pattern Recognition For Human Behaviour (CVPR4HB)*.
- [Battison, 2000] Battison, R. (2000). *Linguistics of American Sign Language: An introduction*, chapter Analyzing Signs, pages 193–212. Gallaudet University Press.
- [Battison et al., 1975] Battison, R., Markowicz, H., and Woodward, J. C. (1975). A good rule of thumb: Variable phonology in American Sign Language. In *Analyzing variation in language*. Washington, DC: Georgetown University Press.
- [Bayley et al., 2002] Bayley, R., Lucas, C., and Rose, M. (2002). Phonological variation in american sign language - the case of 1 handshape. *Language Variation and Change*, 14:19–53.
- [Beal, 2003] Beal, M. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London.
- [Bowden et al., 2004] Bowden, R., Windridge, D., Kadir, T., Zisserman, A., and Brady, M. (2004). A linguistic feature vector for the visual interpretation of sign language. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [Bray et al., 2007] Bray, M., Koller-Meier, E., and Van Gool, L. (2007). Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding (CVIU)*, 106(1):116–129.
- [Brentari, 1998] Brentari, D. (1998). *A Prosodic Model of Sign Language Phonology*. MIT Press, Cambridge, MA.

- [Buehler et al., 2008] Buehler, P., Everingham, M., Huttenlocher, D., and Zisserman, A. (2008). Long term arm and hand tracking for continuous sign language TV broadcasts. In *Proceedings of British Machine Vision Conference (BMVC)*.
- [Buehler et al., 2009] Buehler, P., Everingham, M., and Zisserman, A. (2009). Learning sign language by watching TV (using weakly aligned subtitles). In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Chang et al., 2005] Chang, W. Y., Chen, C. S., and Hung, Y. P. (2005). Appearance-guided particle filtering for articulated hand tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Cooper et al., 2011] Cooper, H., Holt, B., and Bowden, R. (2011). Sign language recognition. In Moeslund, T. B., Hilton, A., Krger, V., and Sigal, L., editors, *Visual Analysis of Humans*, pages 539–562. Springer London.
- [Crasborn et al., 2012] Crasborn, O., Zwitserlood, I., and Ros, J. (2012). Corpus NGT. an open access digital corpus of movies with annotations of sign language of the netherlands <http://www.ru.nl/corpusngtuk/>.
- [Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [de La Gorce et al., 2008] de La Gorce, M., Paragios, N., and Fleet, D. J. (2008). Model-based hand tracking with texture, shading and self-occlusions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Demey and Van der Kooij, 2008] Demey, E. and Van der Kooij, E. (2008). Phonological patterns in a dependency model: Allophonic relations grounded in phonetic and iconic motivation. *Lingua*, 118:11091138.
- [Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38. With discussion.
- [Ding and Martinez, 2009] Ding, L. and Martinez, A. M. (2009). Modelling and recognition of the linguistic components in american sign language. *Image and Vision Computing (IVC)*, 27 (12):1826 – 1844.
- [Dreuw et al., 2006] Dreuw, P., Deselaers, T., Rybach, D., Keysers, D., and Ney, H. (2006). Tracking using dynamic programming for appearance-based sign language recognition. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 293 –298.
- [Dreuw and Ney., 2008] Dreuw, P. and Ney., H. (2008). Visual modeling and feature adaptation in sign language recognition. In *ITG Conference on Speech Communication*.

- [Erol et al., 2007] Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., and Twombly, X. (2007). Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding (CVIU)*, 108:52–73.
- [Farhadi et al., 2007] Farhadi, A., Forsyth, D., and White, R. (2007). Transfer learning in sign language. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Felzenszwalb and Zabih, 2011] Felzenszwalb, P. and Zabih, R. (2011). Dynamic programming and graph algorithms in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 (4).
- [Felzenszwalb et al., 2010] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645.
- [Fillbrandt et al., 2003] Fillbrandt, H., Akyol, S., and Kraiss, K. F. (2003). Extraction of 3D hand shape and posture from image sequences for sign language recognition.
- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381 – 395.
- [Fujimura and Liu, 2006] Fujimura, K. and Liu, X. (2006). Sign recognition using depth image streams. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 381 –386.
- [Green, 1995] Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732.
- [Han et al., 2009] Han, J., Awad, G., and Sutherland, A. (2009). Modelling and segmenting subunits for sign language recognition based on hand motion analysis. *Pattern Recognition Letters*, 30(6):623 – 633.
- [Hanke et al., 2012] Hanke, T., Knig, S., Konrad, R., Langer, G., and Rathmann, C. (2012). German Sign Language corpus (DGS-corpus) <http://www.sign-lang.uni-hamburg.de/dgs-korpus/>.
- [Heap and Hogg, 1996] Heap, T. and Hogg, D. (1996). Towards 3D hand tracking using a deformable model. pages 140–145.
- [Huang et al., 2006] Huang, X., Paragios, N., and Metaxas, D. N. (2006). Shape registration in implicit spaces using information theory and free form deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1303–1318.
- [Israel and Sandler, 2009] Israel, A. and Sandler, W. (2009). Phonological category resolution: A study of handshapes in younger and older sign languages. *Cadernos de Sade, Special Issue Lnguas Gestuais*, 2:13–28.

- [Jelinek, 1997] Jelinek, F. (1997). *Statistical methods for speech recognition*. The MIT Press.
- [Johnston, 2012] Johnston, T. (2012). Auslan Signbank <http://www.auslan.org.au/dictionary/>.
- [Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic models for segmenting and labelling sequence data. In *Proceedings of International Conference on Machine Learning (ICML)*.
- [Liddell and Johnson, 1995] Liddell, S. and Johnson, R. (1995). American Sign Language: The phonological base. In *Linguistics of American Sign Language. An Introduction*. Gallaudet University Press, Washington, DC, (first published in 1989, Sign Language Studies 64, 195277).
- [Liu et al., 2008] Liu, C., Yuen, J., Torralba, A., Sivic, J., and Freeman, W. T. (2008). SIFT flow: Dense correspondence across different scenes. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [Liwicki and Everingham, 2009] Liwicki, S. and Everingham, M. (2009). Automatic recognition of fingerspelled words in british sign language. In *Proceedings of Workshop on Computer Vision and Pattern Recognition For Human Behaviour (CVPR4HB)*.
- [Ljolje and Levinson, 1991] Ljolje, A. and Levinson, S. (1991). Development of an acoustic-phonetic Hidden Markov Model for continuous speech recognition. *IEEE Transactions on Signal Processing*, 39(1):29–39.
- [Lu et al., 2003] Lu, S., Metaxas, D., and Samaras, D. (2003). Using multiple cues for hand tracking and model refinement. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 443–450.
- [Morency et al., 2007] Morency, L.-P., Quattoni, A., and Darrell, T. (2007). Latent-dynamic discriminative models for continuous gesture recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Neidle, 2011] Neidle, C. (2011). Movies of handshapes used in American Sign Language from different views, <http://www.bu.edu/asllrp/cslgr/pages/ncslgr-handshapes.html>.
- [Neidle, 2012] Neidle, C. (2012). National center for sign language and gesture resources <http://www.bu.edu/asllrp/cslgr/>.
- [Neidle, 2013] Neidle, C. (2013). American Sign Language linguistic research project (ASLLRP), <http://www.bu.edu/asllrp/>.
- [Neidle, 2007] Neidle, C. (Reports No. 11 (2002) and 13 (addendum, 2007)). SignStream annotation: Conventions used for the American Sign Language Linguistic Research Project. Technical report, Boston University.

- [Neidle et al., 2012a] Neidle, C., Sclaroff, S., and Athitsos, V. (2012a). American Sign Language Lexicon Video Dataset <http://www.bu.edu/asllrp/lexicon/>.
- [Neidle et al., 2012b] Neidle, C., Thangali, A., and Sclaroff, S. (2012b). Challenges in development of the American Sign Language Lexicon Video Dataset (ASLLVD) corpus. In *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC*.
- [Oikonomidis et al., 2011] Oikonomidis, I., Kyriazis, N., and Argyros, A. (2011). Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Oikonomidis et al., 2012] Oikonomidis, I., Kyriazis, N., and Argyros, A. (2012). Tracking the articulated motion of two strongly interacting hands. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Ong and Bowden, 2004] Ong, E. J. and Bowden, R. (2004). A boosted classifier tree for hand shape detection.
- [Pavlovic et al., 1997] Pavlovic, V. I., Sharma, R., and Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:677–695.
- [Pilet et al., 2008] Pilet, J., Lepetit, V., and Fua, P. (2008). Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision (IJCV)*, 76(2):109–122.
- [Pitsikalis et al., 2011] Pitsikalis, V., Theodorakis, S., Vogler, C., and Maragos, P. (2011). Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. In *Computer Vision and Pattern Recognition Workshops*, pages 1–6.
- [Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press.
- [Roussos et al., 2010] Roussos, A., Theodorakis, S., Pitsikalis, V., and Maragos, P. (2010). Hand tracking and affine shape-appearance handshape sub-units in continuous sign language recognition. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- [Schembri, 2012] Schembri, A. (2012). British Sign Language corpus project <http://www.bsllrp.org/>.
- [Shotton et al., 2011] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [Starner et al., 1998] Starner, T., Weaver, J., and Pentland, A. (1998). Real-time american sign language recognition using desk- and wearable computer-based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375.
- [Stenger et al., 2001] Stenger, B., Mendonca, P. R. S., and Cipolla, R. (2001). Model-based 3D tracking of an articulated hand. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Sudderth et al., 2004] Sudderth, E. B., Mandel, M. I., Freeman, W. T., and Willsky, A. S. (2004). Visual hand tracking using nonparametric belief propagation. In *Proceedings of Workshop on Generative Model Based Vision, CVPR*.
- [Sutton et al., 2004] Sutton, C., Rohanimanesh, K., and McCallum, A. (2004). Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *Proceedings of International Conference on Machine Learning (ICML)*.
- [Teh, 2006] Teh, Y. W. (2006). A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992. Association for Computational Linguistics.
- [Thangali et al., 2011] Thangali, A., Nash, J., Sclaroff, S., and Neidle, C. (2011). Exploiting phonological constraints for handshape inference in ASL video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Thangali and Sclaroff, 2009] Thangali, A. and Sclaroff, S. (2009). An alignment based similarity measure for hand detection in cluttered sign language video. In *Proceedings of Workshop on Computer Vision and Pattern Recognition For Human Behaviour (CVPR4HB)*.
- [Tomasi et al., 2003] Tomasi, C., Petrov, S., and Sastry, A. (2003). 3D tracking = classification + interpolation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.
- [Valli, 2005] Valli, C. (2005). *The Gallaudet Dictionary of American Sign Language*. Gallaudet University Press.
- [Valli and Lucas, 2000] Valli, C. and Lucas, C. (2000). *Linguistics of American Sign Language: An Introduction*. Gallaudet University Press.
- [Van der Kooij, 2002] Van der Kooij, E. (2002). *Phonological Categories in Sign Language of the Netherlands: The Role of Phonetic Implementation and Iconicity*. PhD thesis, Leiden University.
- [Vogler and Metaxas, 2001] Vogler, C. and Metaxas, D. (2001). A framework for recognizing the simultaneous aspects of American Sign Language. *Computer Vision and Image Understanding (CVIU)*, 81:358–384.



- [Vogler and Metaxas, 2004] Vogler, C. and Metaxas, D. (2004). Handshapes and movements: Multiple-channel ASL recognition. In *Proceedings of the Gesture Workshop '03, Genova, Italy*, volume 2915, pages 247–58. Springer Lecture Notes in Artificial Intelligence.
- [von Agris et al., 2007] von Agris, U., Zieren, J., Canzler, U., Bauer, B., and Kraiss, K.-F. (2007). Recent developments in visual sign language recognition. *Universal Access in the Information Society*, 6 (4):323–362.
- [Wang et al., 2008] Wang, J., Athitsos, V., Sclaroff, S., and Betke, M. (2008). Detecting objects of variable shape structure with hidden state shape models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):477–492.
- [Wang et al., 2006] Wang, S., Quattoni, A., Morency, L., Demirdjian, D., and Darrell, T. (2006). Hidden conditional random fields for gesture recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [Weiss et al., 2010] Weiss, D., Sapp, B., and Taskar, B. (2010). Sidestepping intractable inference with structured ensemble cascades. *Advances in Neural Information Processing Systems*, 23:2415–2423.
- [Whitworth, 2011] Whitworth, C. F. (2011). *Features, clusters, and configurations: Units of contrast in American Sign Language handshapes*. PhD thesis, Gallaudet University.
- [Wu et al., 2005] Wu, Y., Lin, J., and Huang, T. S. (2005). Analyzing and capturing articulated hand motion in image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1910–1922.
- [Yang et al., 2009] Yang, H.-D., Sclaroff, S., and Lee, S.-W. (2009). Sign language spotting with a threshold model based on conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1264–1277.
- [Yang et al., 2010] Yang, R., Sarkar, S., and Loeding, B. (2010). Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):462–477.
- [Yin, 2010] Yin, P. (2010). *Segmental discriminative analysis for American Sign Language recognition and verification*. PhD thesis, Georgia Institute of Technology.
- [Yin et al., 2009] Yin, P., Starner, T., Hamilton, H., Essa, I., and Rehg, J. (2009). Learning the basic units in american sign language using discriminative segmental feature selection. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4757–4760.



Ashwin Thangali  
Curriculum Vitae  
April 2013

**Personal Info**

Citizenship: India  
Languages: English, Hindi, Kannada

**Contact Info**

Boston University  
Department of Computer Science  
111 Cummington St.  
Boston, MA 02215  
e-mail: tvashwin@cs.bu.edu  
web: <http://cs-people.bu.edu/tvashwin>

**Education**

- Boston University, Computer Science Dept.: Ph.D. (2013 expected)
- Indian Institute of Science, Bangalore, Electrical Engineering: M.E. (System Science and Automation, 2000)
- National Institute of Technology Karnataka, Surathkal: B.E. (1998)

**Professional Appointments**

- 2003 – present: Research Assistant and Teaching Fellow, CS Dept., Boston University
- Summer 2012: Research internship, Image Analytics Lab, GE Global Research Center, Niskayuna, NY
- 2000 – 2003: Research Staff Member, IBM Research, New Delhi, India

**Academic awards**

- Best teaching fellow award, Computer Science, 2004
- Hariri Award for Innovative Computing Models, Algorithms, and Systems Research, April 2013

**Professional activities**

- Served as reviewer for CIVR, CVPR, ECCV, ICCV, IJCV, CVIU and Transactions on Image Processing
- Served as co-organizer of IVC weekly research colloquiums during 2009-2010

**Publications****Publications in refereed journals**

1. Quan Yuan, Ashwin Thangali, Vitaly Ablavsky and Stan Sclaroff, "Learning a Family of Detectors via Multiplicative Kernels", *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, Vol. 33(3), 2011.
2. Gaurav Aggarwal, Ashwin Thangali and Sugata Ghosal, "An image retrieval system with automatic query modification", *IEEE Transactions on Multimedia*, 4(2): 201-214, 2002.
3. Ashwin Thangali and P.S. Sastry, "Font and Size Independent OCR System for Printed Kannada Documents using Support Vector Machines", *Sadhana*, Vol.27, pp.35-58, Feb. 2002

#### **Publications in refereed proceedings**

1. Zheng Wu, Ashwin Thangali, Stan Sclaroff, Margrit Betke, "Coupling Detection and Data Association for Multiple Object Tracking", *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
2. Carol Neidle, Ashwin Thangali, Stan Sclaroff, "Challenges in development of the American Sign Language Lexicon Video Dataset (ASLLVD) corpus", 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC 2012.
3. Ashwin Thangali, Joan Nash, Stan Sclaroff and Carol Neidle, "Exploiting Phonological Constraints for Handshape Inference in ASL Video", In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
4. Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Ashwin Thangali, Haijing Wang and Quan Yuan, "Large Lexicon Project: American Sign Language Video Corpus and Sign Language Indexing/Retrieval Algorithms", In *Proc. Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora*, 2010.
5. Ashwin Thangali and Stan Sclaroff, "An alignment based similarity measure for hand detection in cluttered sign language video", In *Proc. IEEE Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis (CVPR4HB)*, 2009.
6. Vitaly Ablavsky, Ashwin Thangali and Stan Sclaroff, "Layered graphical models for tracking partially-occluded objects", In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
7. Quan Yuan, Ashwin Thangali, Vitaly Ablavsky and Stan Sclaroff, "Multiplicative Kernels: Object Detection, Segmentation and Pose Estimation", In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
8. Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Quan Yuan and Ashwin Thangali, "The American Sign Language Lexicon Video Dataset", In *Proc. IEEE Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis (CVPR4HB)*, 2008.

9. Quan Yuan, Ashwin Thangali, Vitaly Ablavsky and Stan Sclaroff, "Parameter Sensitive Detectors", In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.
10. Daniel Gutchess, Vitaly Ablavsky, Ashwin Thangali, Stan Sclaroff and Magnus Snorrason, "Video Surveillance of Pedestrians and Vehicles", In Proc. SPIE Conference on Tracking, Pointing and Laser Systems Technologies XXI, Vol. 6569, 2007.
11. Quan Yuan, Ashwin Thangali and Stan Sclaroff, "Face Identification by a Cascade of Rejection Classifiers", In Proc. IEEE Workshop on Face Recognition Grand Challenge Experiments, 2005.
12. Ashwin Thangali and Stan Sclaroff, "Periodic Motion Detection and Estimation via Space-Time Sampling", In Proc. IEEE Workshop on Motion and Video Computing, 2005.
13. Ashwin Thangali, Rahul Gupta and Sugata Ghosal, "Leveraging Non-Relevant Images To Enhance Image Retrieval Performance", In Proc. ACM Conference on Multimedia 2002.
14. Ashwin Thangali, Rahul Gupta and Sugata Ghosal, "Adaptable Similarity Search using Non-Relevant Information", 28th International Conference on Very Large Data Bases (VLDB) 2002.
15. Adams et.al., "IBM Research TREC-2002 Video Retrieval System", Text Retrieval Conference (TREC) 2002, Video Retrieval Track.
16. Ashwin Thangali, Sugata Ghosal and Navendu Jain, "Improving image retrieval performance with Negative relevance Feedback", International Conference on Acoustics Speech and Signal Processing (ICASSP) 2001.

### **Teaching Experience**

#### **Teaching Fellow for:**

- CS480/680 (Fall 2003): Introduction to Computer Graphics
- CS460/660 (Spring 2004): Introduction to Database Systems
- CS112: Introduction to Data Structures and Algorithms
- CS111: Introduction to Computer Science I

#### **Research Interns Supervised at Boston University:**

- Tianxiong Jiang (B.A., Computer Science, Boston University)
- Eric Cornelius (B.A., Computer Science, Boston University)