Representation and Learning for Sign Language Recognition

Sunita Nayak
University of South Florida

Follow this and additional works at: https://scholarcommons.usf.edu/etd
Part of the American Studies Commons

Scholar Commons Citation

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
Representation and Learning for Sign Language Recognition

by

Sunita Nayak

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Sudeep Sarkar, Ph.D.
Arthur I. Karshmer, Ph.D.
Barbara Loeding, Ph.D.
Dmitry Goldgof, Ph.D.
Rangachar Kasturi, Ph.D.

Date of Approval:
January 17, 2008

Keywords: American Sign Language, gestures, human-human interaction, probabilistic distances, motion, learning common motion patterns

© Copyright 2008, Sunita Nayak
DEDICATION

To my husband, Satya,
and my parents, brother and sister !!
ACKNOWLEDGEMENTS

I thank my major professor Prof. Sudeep Sarkar for his immense patience, co-operation and great guidance throughout my Ph.D. program. I also thank my committee member Prof. Barbara Loeding for her continuous guidance throughout my graduate studies, and Prof. Rangachar Kasturi, Prof. Dmitry Goldgof and Prof. Arthur Karshmer for reviewing my dissertation and their useful comments. I could not have completed my Ph.D. without the support and patience of my husband. Thanks to the Tech Support group of the Computer Science department at USF for their prompt help anytime I had a technical problem. Many thanks to my lab-mates in the vision lab for sharing their knowledge and helping me out whenever needed.
TABLE OF CONTENTS

LIST OF TABLES iii
LIST OF FIGURES iv
ABSTRACT vi

CHAPTER 1 INTRODUCTION 1
1.1 Motivation 1
1.2 The Need for Embedding 3
1.3 Automated Learning of Models 4
1.4 Contributions 7
1.5 Layout of the Dissertation 8

CHAPTER 2 RELATED WORK 9
2.1 Sign Language Recognition 10
2.2 Common Pattern Extraction 12

CHAPTER 3 EMBEDDING PROBABILITY FUNCTIONS IN LOW DIMENSIONAL CONFIGURATION SPACES 15
3.1 Relational Distributions 15
3.2 Dimensionality Reduction of Relational Distributions 17
3.3 Embedding New Samples 23
3.4 Choice of Reference Points 26
3.5 Time Complexity 27
3.6 Matching 27
3.7 Results and Analyses 28
  3.7.1 Datasets 29
    3.7.1.1 Signs from American Sign Language 29
    3.7.1.2 Two-Handed Gestures 30
    3.7.1.3 Human-Human Interaction 30
3.7.2 Recognition Performance 31
3.7.3 Choice of Number of Reference Points 35
3.7.4 Effect of the Dimensions of the Embedding 35
LIST OF TABLES

Table 2.1  Sign language recognition approaches  11

Table 3.1  Probabilistic distance measures between two probability distributions along with their expressions in terms of inner products (last column) between transformed versions of the probability functions.  19

Table 3.2  Recognition performance on the ASL dataset with 147 different test signs.  32

Table 3.3  Recognition performance on the two-handed test gesture sequences based on 98 test sequences.  32

Table 3.4  Recognition performance on the human interaction dataset for 8 test sequences.  33

Table 5.1  Notations used in this chapter  55
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Variations in the relational distributions with motion.</td>
<td>2</td>
</tr>
<tr>
<td>Figure 1.2</td>
<td>Movement epenthesis in sign language sentences.</td>
<td>6</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Overview of the approach along with samples of the intermediate representations.</td>
<td>20</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Embedding probability density functions.</td>
<td>36</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Five ASL signs.</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Seven different types of two-handed gestures.</td>
<td>38</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>Eight different types of human interaction sequences.</td>
<td>39</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Extracted contours for the human-human interaction sequences.</td>
<td>40</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>Effect of the number of reference points used during embedding.</td>
<td>41</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>Effect of the percentage of energy retained with various distance measures.</td>
<td>42</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Example of relational distributions constructed using exhaustive and random sampling.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Effect of using sampling methods to estimate the relational distributions.</td>
<td>45</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Illustration of robustness of relational distributions with respect to parameters.</td>
<td>46</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Example of interpolation in configuration space.</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Recognition results on the two-handed gesture dataset at various temporal scales</td>
<td>51</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Concept of signemes.</td>
<td>53</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Overview of our approach for signeme extraction.</td>
<td>54</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.3</td>
<td>Convergence of values of the parameter set.</td>
<td>61</td>
</tr>
<tr>
<td>5.4</td>
<td>Convergence of the conditional probability density.</td>
<td>62</td>
</tr>
<tr>
<td>5.5</td>
<td>Histograms showing the start and end locations of signs extracted from 14 different sentences using multiple ICM runs.</td>
<td>64</td>
</tr>
<tr>
<td>5.6</td>
<td>First dimension of the video sequences containing a common sign ‘DEPART’.</td>
<td>68</td>
</tr>
<tr>
<td>5.7</td>
<td>Extraction of common patterns from 136 video sequences.</td>
<td>69</td>
</tr>
<tr>
<td>5.8</td>
<td>Extraction of common patterns from 136 audio sequences.</td>
<td>69</td>
</tr>
<tr>
<td>5.9</td>
<td>Signemes extracted from sentences</td>
<td>70</td>
</tr>
<tr>
<td>5.10</td>
<td>Signemes extracted from sentences</td>
<td>71</td>
</tr>
<tr>
<td>5.11</td>
<td>Some partially correct and bad extractions</td>
<td>71</td>
</tr>
<tr>
<td>A.1</td>
<td>A screen shot of the webpage showing the common pattern extraction results.</td>
<td>85</td>
</tr>
</tbody>
</table>
While recognizing some kinds of human motion patterns requires detailed feature representation and tracking, many of them can be recognized using global features. The global configuration or structure of an object in a frame can be expressed as a probability density function constructed using relational attributes between low-level features, e.g. edge pixels that are extracted from the regions of interest. The probability density changes with motion, tracing a trajectory in the latent space of distributions, which we call the configuration space. These trajectories can then be used for recognition using standard techniques such as dynamic time warping. Can these frame-wise probability functions, which usually have high dimensionality, be embedded into a low-dimensional space so that we can still estimate various meaningful probabilistic distances in the new space? Given these trajectory-based representations, can one learn models of signs in an unsupervised manner? We address these two fundamental questions in this dissertation.

Existing embedding approaches do not extend easily to preserve meaningful probabilistic distances between the samples. We present an embedding framework to preserve the probabilistic distances like Chernoff, Bhattacharya, Matusita, KL or symmetric-KL based on dot-products between points in this space. It results in computational savings. We experiment with the five different probabilistic distance measures and show the usefulness of the representation in three different contexts - sign recognition of 147 different signs (with
large number of possible classes), gesture recognition with 7 different gestures performed by 7 different persons (with person variations) and classification of 8 different kinds of human-human interaction sequences (with segmentation problems).

Currently, researchers in continuous sign language recognition assume that the training signs are already available and often those are manually selected from continuous sentences. It consumes a lot of human time and is tedious. We present an approach for automatically learning signs from multiple sentences by using a probabilistic framework to extract the parts of signs that are present in most of its occurrences, and are robust to variations produced by adjacent signs. We show results by learning 10 signs and 10 spoken words from 136 sign language sentences and 136 spoken sequences respectively.
1.1 Motivation

Articulated motion analysis is fundamental to sign language recognition, gesture recognition or human activity classification. Reviews of work in articulated human motion appear in [1–4]. While some of the existing approaches are based on magnetic or optical markers [5–8], some are based on pure image-based input approaches [9–12]. Some use parametric models [13–17] for representation, while some are non-model based approaches [18]. Further, some approaches to articulated human motion recognition include those that rely on body part labeling and tracking [13, 17, 19, 20] and those that do not [21–23]. The latter class of approaches does not require part tracking or correspondence between the frames of video sequence, and do not rely on geometric object models like skeletons.

Our work is concerned with those tracking-free, image-based, parametric model-free approaches that abstract each frame as a distribution of low-level features such as edge orientation, flow distribution, or relations between features. For instance, orientation histogram has been used in gesture recognition [24] to represent the hand in images of a sequence. Articulated motion involves change in relative configuration of the parts, which manifests itself as evolution of these distributions. Figure 1.1 depicts how motion can be captured in terms of distributions. The first column shows a motion sequence, the second column shows the edge pixels in skin color blobs, and the third column shows top view of
the distributions constructed from the low level features (edge pixels) of the images in the motion sequence. The horizontal axis of the distribution represents the horizontal distance between the edge pixels and its vertical axis represents the vertical distance between the edge pixels. Note the change in the distributions as the signer moves her hand.

The advantage of using distribution-based representations is that it does not need part-level tracking nor does it require that the segmentation be very precise; it is tolerant to fragmentation of boundary and some level of spurious features. Such distribution-based ideas have also been recently proposed for the object detection problem in images. For instances, spatial-color joint distribution has been used in [25] and joint statistics are used for human detection in [26]. Color distributions or histograms are routinely used in image-database indexing. In this dissertation, we demonstrate our ideas using a global statistical representation, called the relational distribution, to represent the global part configuration.
structure in each image frame in a Gestaltic manner. In sign language recognition, there is experimental evidence that signs can be recognized from point light displays [27, 28], emphasizing that for a broad class of signs just global configuration of hands and face is sufficient for recognition. The relational distribution is in itself a probabilistic density function using a histogram representation. It uses distribution of pairwise relationships between the low-level primitives, such as edge points, and has been shown to be useful for human motion recognition [21] and sign language recognition [29].

1.2 The Need for Embedding

These distributions, in particular the relational distributions, have high dimensionality, with the number of dimensions being equal to the total number of bins in the histogram. Apart from the discretization parameter, the number of bins in the histogram is also proportional to the number of attributes one uses to describe the relationship between two low-level primitives. Although we have used just the displacement between two edge points as the attributes resulting in two dimensions, this is not the only possible choice. One could also think of using the local gradient information such as gradient magnitude or the direction as two more attributes. Other possibilities include using optic flow vector information at the edge point, if available. With the addition of new attributes, the relational distribution gets richer, however, the size of the histogram representation increases exponentially. Thus, there is a need for a low-dimensional embedding of these relational distributions. Existing embedding methods include Principal Component Analysis(PCA), kernelPCA [30], multi-dimensional scaling [31], isomap [32], local-linear embedding [33] and Laplacian eigenmaps [34]. However, unlike these approaches, we do not seek to directly match the distances in the embedding space with the probabilistic distances between relational distributions; this will not be possible for many non-linear probabilistic distance measures.
Rather we propose a novel strategy that allows us to construct spaces that allow the indirect estimation of the probabilistic distances.

We express each probability distance measure as a sum of dot products of the transformed functions, where the transformations are decided based on the nature of the specific distance measure. The transformed functions are then embedded into a low-dimensional space by performing a Singular Value Decomposition, such that their pair-wise dot products are preserved in the new space. This results in computational savings. We will refer to this low-dimensional embedding space of the relational distributions as the configuration space. New relational distributions are embedded into the space by retaining the distance of the new transformed probability functions from a subset of all the training transformed probabilities using an iterative minimization scheme. We choose the elements in the subset by performing uniform random sampling on the set of all transformed probabilities in the training set. We show that our approach works for five different probabilistic distance measures: Bhattacharya [35], Matusita [36], KL, Symmetric KL [37], and Chernoff [38] distance measures. We apply the embedding to articulated motion recognition, but the embedding process is a general one that can be used to embed probability functions for other vision applications such as in image databases, where histogram representations are common.

1.3 Automated Learning of Models

Most sign recognition approaches are generative model-based, using HMMs, that require huge amount of training data. But most of them assume that the training signs are already available and often signs used in the training set are isolated signs with their starting and end boundaries manually chopped off, or manually selected set of frames from continuous sentences. In this dissertation, we present an approach to automatically extract the
signs from continuous sentences. While the extraction is based on relational distribution type of representation, the extracted models are at frame level, i.e. in terms of a collection of frames and can be used by other ASL researchers too who might be using other feature types.

While signing a sign language sentence, there exist transition of the hands between two consecutive signs that do not belong to either of the signs. This is called movement epenthesis [39]. This needs to be dealt first before dealing with any other phonological issues in sign language [40]. Figure 1.2 shows two sentences ‘I BUY TICKET WHERE?’ and ‘YOU CAN BUY THIS FOR HER’ with a common sign ‘BUY’ between them that is marked in red. The adjacent signs in each sentence are marked in magenta. The unmarked frames between the signs indicate the frames corresponding to movement epenthesis. It can be observed that the same sign ‘BUY’ is preceded and succeeded by movement epenthesis that depends on the end and start of the preceding and succeeding sign respectively. The movement epenthesis also effects how the sign is signed. This effect makes the automated extraction, learning and recognition of signs from continuous sentences more difficult when compared to just plain gestures, isolated signs or fingerspelling.

Given a set of sentences with only one common sign, we automatically extract the common sign from each of the sentences. Each sentence is represented as a sequence of points in a low-dimensional space arrived at by performing PCA on all relational distributions from the sentences. The starting location and width of the common pattern in each of the sentences, together form a parameter set. The parameter set is initialized using uniform random sampling and for each initialization, it is optimized using Iterated Conditional Modes (ICM). At each ICM iteration, the starting location and width values of the common pattern in a sentence are updated based on the locations and widths of the common pattern in all other sentences. The parameter values are iteratively updated till convergence. Given the local convergent nature of ICM, we repeat it with different initializations multiple num-
ber of times. The number of times ICM is run is decided based on the average length of all the sentences. The most frequently occurring solution from all the ICM runs is considered as the final solution. We call these extracted common patterns from the sentences as signemes. They can be used for spotting or recognition of signs in continuous sign language sentences using either Hidden Markov Models or Dynamic Time warping. Sign language experts can also use the extracted set of frames for teaching or studying variations between instances of signs in continuous sign language sentences, or in automated sign language tutoring systems. The signeme extraction algorithm can also be applied to audio data. The
extracted common words(spoken) can then be used in the process of annotating the video signemes.

1.4 Contributions

The contributions of this dissertation are:

1. We propose a novel framework for efficiently embedding the relational distributions in a low dimensional configuration space preserving the probabilistic distances between them. Relational distributions can be used to study the dynamics involved in motion sequences, detect changes in scenes in videos or classify motion sequences in different domains.

2. We study the robustness of relational distributions with respect to varying edge parameters, and present a random sampling-based approach for fast estimation of relational distributions. We also show that interpolation of motion in the configuration space increases recognition accuracy when there are speed variations between various gesture instances.

3. We present recognition results on different kinds of datasets without using any form of tracking or correspondence between frames. We perform recognition of signs from American Sign Language, two-handed gesture recognition, and classification of human-human interaction sequences.

4. We propose an unsupervised algorithm to automatically learn sign models from multiple sign language sentences. Our models are at frame level which makes them usable by other ASL researchers using different low-level features as well.
1.5 Layout of the Dissertation

We organized the dissertation as follows. Chapter 2 presents the related work in the areas of articulated motion in general, those specifically in sign language recognition and those in common pattern extraction. Chapter 3 reviews the concept of relational distributions and describes how we embed the relational distributions, which are probability density functions, into a low dimensional space. It also explains how a new relational distribution corresponding to an image in a test sequence can be projected onto the space. Chapter 4 presents empirical results on the robustness of the relational distributions, gives a random sampling based approach for their faster computation, and describes how motion can be represented as a smooth curve in the configuration space using interpolation. Chapter 5 describes how we extract signemes from multiple sign language sentences simultaneously. Finally, we conclude the dissertation in Chapter 6 by summarizing our findings and discussing possible future work.
CHAPTER 2
RELATED WORK

Our motion representation is essentially state-space based, where instead of the state being specified by the features extracted from the images [10, 21, 41] or by the model parameters [15, 16] or by pixel locations (motion trajectories) [42], it is specified by the configuration space coordinates. In this space, state-based methods such as Hidden Markov Models (HMMs)) [43–45], Conditional Random Fields [46], Maximum Entropy Markov Models [47] and Hidden State Conditional Random Fields [11] can be used with relational distributions, but they need more training data. Our representation is more close to the statistical continuous curve representation by Bobick and Wilson [48]. Matching of motion state space trajectories is typically done using dynamic programming [43] or trajectory correlation [21] or more recently, using salient points on the trajectories [49]. There has also been work recently in aligning motion sequences [50].

Our work does not use tracking. A related approach is Bobick and Davis’s representation [51] of human motion in short video sequences using motion history images. On conceptually similar grounds is Masoud and Papanikolopoulos’s [52] feature images used to represent motion in an image sequence without using limb tracking. Whole body contours have also been used for gesture recognition [41]. Zahedi et al. [53] used appearance based models for sign recognition. Spatio-temporal shapes have been used recently [18, 23, 50] for action recognition. We use relational distributions because of their wide applicability across various domains [21, 29] and the flexibility to incorporate higher discrimination.
power into them whenever needed. They also allow us to interpolate motion, if required, for datasets with low frame capture rate or while matching motions with varying speeds.

2.1 Sign Language Recognition

A major portion of this dissertation is on sign language representation and recognition. Ong and Ranganath [40] and Loeding et al. [54] present reviews on the automated sign language research. Table 2.1 lists some of the selected works in sign language recognition with their vocabulary size and recognition rate. Many researchers in sign language overcome the difficult problem of tracking hands by using magnetic trackers or Cyber Gloves [5, 55–57]. Some systems used skin color detection to segment out the hands and face [42, 58–63]. They required the signer to wear long sleeves and restricted the presence of any skin color object in the background. Some systems assumed that hand is the only moving object in the scene [64–67]. Some also assumed that they move constantly and they were detected from the differences of consecutive images [64, 65, 68]. Assan and Grobel [69], Bauer and Kraiss [70, 71], and Tamura and S. Kawasaki [72] tried to use the geometry of the whole body and localized the body torso. Hienz et al. [73] also localized the elbow and shoulders, along with the hands and face. Bowden et al. [74] used a classifier bank of Markov chains for detecting the hands in the images.

Tracking of the hands in 2D images has been tried by many researchers in sign language recognition. It is done by using color-based blob extraction in the images [59, 62, 80], using contours or boundary models [64, 66, 68] or by using motion cues [42]. But tracking becomes very difficult when the two hands occlude each other or when they come infront of the face that has the same skin color as the hands. While some use color gloves as a simple solution to it [69–71, 84, 85], some used different viewpoints to prevent occlusions [62].
Table 2.1. Sign language recognition approaches

<table>
<thead>
<tr>
<th>Researchers</th>
<th>Classification Methods</th>
<th>Vocab.</th>
<th>Rec. Rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approaches using direct measuring devices such as magnetic trackers or data gloves</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fang et al. [75]</td>
<td>HMM, Self-organizing map, recurrent NN</td>
<td>208 CSL</td>
<td>92.1</td>
</tr>
<tr>
<td>Kadous [76]</td>
<td>Instance-based learning</td>
<td>95 Auslan</td>
<td>80</td>
</tr>
<tr>
<td>Murakami &amp; Taguchi [77]</td>
<td>Recurrent Neural Networks</td>
<td>10 JSL</td>
<td>96</td>
</tr>
<tr>
<td>Wang et al. [5]</td>
<td>HMM model sequential subunits</td>
<td>5119 CSL</td>
<td>92.8</td>
</tr>
<tr>
<td>Wu &amp; Gao [78]</td>
<td>Dynamic Gaussian Mixture Models</td>
<td>274 CSL</td>
<td>97.4</td>
</tr>
<tr>
<td></td>
<td>Vision-based approaches</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assan &amp; Grobel [69]</td>
<td>HMM, used color gloves</td>
<td>262 SLN</td>
<td>91.3</td>
</tr>
<tr>
<td>Bauer &amp; Kraiss [71]</td>
<td>HMM-based subunits, used color gloves</td>
<td>100 GSL</td>
<td>92.5</td>
</tr>
<tr>
<td>Cui &amp; Weng [65]</td>
<td>Recursive PCA, motion-based hand segmentation</td>
<td>28 ASL</td>
<td>93.2</td>
</tr>
<tr>
<td>Huang &amp; Huang [66]</td>
<td>3D Hopfield NN model-based hand tracking</td>
<td>15 TSL</td>
<td>96</td>
</tr>
<tr>
<td>Matsuo et al. [79]</td>
<td>Rule-based, used gloves and stereo</td>
<td>38 JSL</td>
<td>76-79</td>
</tr>
<tr>
<td>Starner et al. [62]</td>
<td>HMM, color cameras at angular views</td>
<td>40 ASL</td>
<td>92-98</td>
</tr>
<tr>
<td>Tanibata et al. [80]</td>
<td>HMM, used only correctly extracted face and hands</td>
<td>65 JSL</td>
<td>100</td>
</tr>
<tr>
<td>Yang et al. [42]</td>
<td>NN, correspondence between consecutive frames</td>
<td>40 ASL</td>
<td>96.2</td>
</tr>
<tr>
<td>Imagawa et al. [60]</td>
<td>PCA+clustering, hand tracking</td>
<td>33 JSL</td>
<td>72-94</td>
</tr>
<tr>
<td>Yang et al. [81]</td>
<td>Relational Histograms+PCA</td>
<td>39 ASL</td>
<td>83</td>
</tr>
<tr>
<td>Parashar [82]</td>
<td>Relational Histograms+PCA</td>
<td>39 ASL</td>
<td>88-95</td>
</tr>
<tr>
<td>Our work [83]</td>
<td>Relational Histograms+Probabilistic distance preserving embedding</td>
<td>147 ASL</td>
<td>80.3</td>
</tr>
</tbody>
</table>

CSL-Chinese Sign Language, SLN-Sign Language of Netherlands, GSL-German Sign Language, TSL-Taiwanese Sign Language, JSL-Japanese Sign Language, NN-Neural Networks
Our approach relies on skin segmentation. It does not need the localization of the body parts other than the hands and the face in the images. We do not perform any kind of tracking of hands or correspondence of salient points between consecutive images. Instead we exploit the relative configuration of the two hands and face in each image and model motion based on change in their relative configuration.

Following the success of Hidden Markov Models (HMMs) in speech recognition, they were used by sign language researchers [62, 74, 86–88] for representing and recognizing signs. But HMMs need a large number of training data and unlike speech, data from native signers is not as easily available as speech data. Hence recently non-HMM based approaches have started being used [29, 89, 90]. Our method uses instance-based learning and hence does not need a large training dataset.

2.2 Common Pattern Extraction

Continuous sign language involves problems that are unseen in isolated sign recognition. As signs are signed continuously in a sentence, adjacent signs effect each other and there exist transition periods where the hands move from end of one sign to the start of the next sign. These transitions are called movement epenthesis. Ranganath [40] discusses these issues in continuous sign language. These are very smooth and swift movements, especially when signed by native signers. They also effect the signs themselves to some extent. Hence the training signs for continuous sign language recognition need to be extracted from the sentences themselves. Most of the existing work in sign language assume that the training signs are already available and often signs used in the training set are the manually selected frames from continuous sentences. This process consumes a lot of human time.
We are the first ones to formulate the sign modeling problem in this manner [29]. Our earlier approaches were based on local searches based on pairs of sentences. In this dissertation, we will present a more elegant solution that adopts more global approach. Another related work is by Bauer and Kraiss [71]. Based on the use of phonemes for speech recognition, they demonstrate extraction of self-organizing subunits from sentences. We demonstrate our approach by extracting signemes, which correspond to the whole signs. But our approach could also be used for extracting the sub-units of signs.

A different but a closely related problem is the extraction of common subsequences, also called motifs, from very long multiple gene sequences in biology [91–94]. Lawence et al. [92] used Gibbs sampling based on discrete match or mis-match of subsequences that were strings of symbols of gene sequences, Bailey and Elkan [91] use expectation maximization to find common subsequences in univariate biopolymer sequences. But in biology, researchers deal with univariate discrete sequences, and hence their algorithms are not always directly applicable to other multivariate continuous domains in time series like speech or sign language. Some time-series researchers tried to symbolize the time series into discrete sequences and use the existing algorithms from bioinformatics. For example, Chiu et al. [95] symbolized the time series into sequence of symbols using local approximations and used random projections for extracting the common subsequences in noisy data. Tanka et al. [96] extended their work by performing PCA on the multivariate time series data and projecting them into a single dimension and then symbolizing the data into discrete sequences. But it is not always possible to get all the important information in the first principal component alone. Further extending his work, recently, Duchne et al. [97] find recurrent patterns from multivariate discrete data using time series random projections.

Due to the inherent continuous nature of many time series data like gesture and speech, new methods were developed for them without approximating the data to sequence of dis-
crete symbols. Denton [98] used a continuous random-walk noise model to cluster similar substrings. Nayak et al [29] and Minnen et al. [99] use continuous multivariate sequences and use dynamic time warping to find distances between the substrings. Oates [100] and Nayak et al. [29] are among the few works in finding recurrent patterns that address non-uniform sampling of time series. The recurrent pattern extraction approach proposed in this dissertation is based on multivariate continuous time series, uses dynamic time warping to find distances between substrings and it can also cope with the variation in length of the common pattern. To our knowledge, there is no work in sign language, demonstrating the automated extraction of signs from continuous sentences, except for [29].
CHAPTER 3
EMBEDDING PROBABILITY FUNCTIONS IN LOW DIMENSIONAL CONFIGURATION SPACES

We contend that it is possible to discriminate among many articulated motions using just the change in the global configuration or structure of the object. For instance in sign language recognition, it should be possible to discriminate between a number of signs based on just the relative movement of the hands, without the need for detailed information such as shape of the individual fingers. Indeed there is some experimental evidence that sign recognition is possible with point light displays [27, 28]. Similarly, it should be possible to distinguish between aggressive and benign interaction between two individuals without the need for limb tracking, but based on the global postures of the two individuals. Of course, such a solution is not a complete solution to the problem; there will be situations that will need detailed analysis. However, as can be seen later in the text, we found that global structure is sufficient for a broad range of articulated motion recognition and classification. In this chapter, we describe how the global configuration of an object in an image can be captured in the form of a probability density function and how we embed these high dimensional probability functions into a low dimensional configuration space.

3.1 Relational Distributions

How do we capture the global configuration of the object? We start with low-level primitives that are most likely to come from the articulated object. The exact nature of the low-level primitives can vary. Some common choices include edges, salient points, Gabor
filter outputs and so on. We use edges in this dissertation. We start from some level of
segmentation of the object from the scene. The exact steps to achieve this are not new and
can vary from domain to domain. They consist of background subtraction, skin color de-
tection or color-based blob extraction. The uses of these processes are fairly standard ones
and have been used widely in gesture and sign recognition. We then consider the bound-
daries of the postulated object regions along with the edges inside them or just the region
blob boundaries as our low-level image primitives. We capture the global configurations
by considering the relationships between these primitives.

Let there be \( n \) low-level primitives in an image that are represented in a set \( F = \{f_1, \cdots, f_n\} \)
and the relationship among \( k \)-tuple primitives is represented by a set of \( M \) attributes \( a = \{a_1, \cdots, a_M\} \). In our experiments, we use the distance between two contour or edge pixels
in the vertical and horizontal direction \((dx, dy)\) as the relational attributes, i.e. \( k = 2 \) and
\( a = \{dx, dy\} \). The joint probability function \( P(a) \) describes the distribution of the primitives
within an image and captures the shape of the pattern in the image. This probability
is called a relational distribution [21]. It captures the global configuration of the low-level
primitives. In rest of the text, we use the notation \( P(a) \) and \( P \) interchangeably. Both of
them denote a relational distribution.

The rigid or non-rigid motion of the objects in an image sequence will result in change
of the distribution of primitives in the images. Earlier in Fig. 1.1, we showed some ex-
ample variations in the relational distributions with motion. It shows the top view of the
distributions. The region near the center represents the points closer to each other, e.g. the
edge points within the face or within the hand, while farther from the center represent the
farther away points, e.g. the relationship between the edge points of a hand and the face.
Notice the change in the relational distributions as the signer moves one of her hands. To
be able to discriminate symmetrically opposite motion, we maintain the signs (or direc-
tions) of the horizontal and vertical distances between the edge pixels in each ordered pair.
This leads to representing the probability distribution in a four quadrant system. Given that these relational distributions exhibit complicated shapes that are difficult to be modeled readily using a combination of simple shaped distributions such as Gaussian mixtures, we adopt non-parametric histogram-based representation. For better discrimination of the probabilities, while forming the histogram, we do not add counts to the center of the histogram which represents the distance of the edge pixels from itself or very close adjacent pixels. Each bin then counts the pairs of edge pixels between which, the horizontal and vertical distances each lie in some fixed range that depends on the location of the bin in the histogram. In our experiments, for simplicity, we use a fixed size histogram for all the sequences. The above range is then defined using linear mapping between the image size and the histogram size, e.g. image size along the horizontal direction corresponds to half the histogram size in the horizontal direction. One could use histogram bin size optimization techniques for optimizing the histograms for each dataset. But we found the fixed size histograms sufficient for our experiments.

3.2 Dimensionality Reduction of Relational Distributions

Fig 3.1 illustrates our approach. Each image frame is represented by a relational distribution that captures the probability that randomly selected pair of image primitives (e.g. edge primitives) exhibit a certain relationship (e.g. horizontal and vertical displacement). This distribution is estimated by the normalized histogram of the relationships between sampled pairs of edge pixels. We have found that such gross, overall representation works well for a wide range of articulated motion scenarios. The relational distributions, have high dimensionality, with the number of dimensions being equal to the total number of bins in the histogram. Their size increases exponentially if more attributes are added to them to further increase their discriminative power. Hence it is important to reduce their
dimensions before further processing. Since the relational distributions themselves are probability functions, we embed them into a low-dimensional space where meaningful probabilistic distance measures can be computed. This embedding process is one of the core theoretical contributions of this dissertation.

We start with expressing the probabilistic distances in terms of dot products and define two simple transformations of the histograms depending on the nature of the distance measure. The transformed histograms are then embedded into the low dimensional space by preserving pairwise dot products between them. We call this reduced dimensional space the configuration space. The probabilistic distance measures are then computed by computations of dot products in the configuration space. This results in computational savings. We show that the same framework works for five different probabilistic distance measures: Bhattacharya [35], Matusita [36], KL, Symmetric KL [37], and Chernoff [38] distance measures.

We have to be careful while constructing the low-dimensional embedding. It should be possible to compute meaningful distance measures from this space. Table 3.1 lists some of the commonly used distance measures between two probability distributions. A common method for dimensionality reduction is using principal component analysis (PCA), as was done in [21]. However, this type of embedding is appropriate only for computing certain types of distance measures between probability functions such as the Euclidean and Matusita distance measure. Existing commonly used dimensionality reduction techniques include kernelPCA [30], multi-dimensional scaling [31], isomap [32], local-linear embedding [33] and Laplacian eigenmaps [34]. However, unlike these approaches, we do not seek to directly match the distances in the embedding space with the probabilistic distances; this will not be possible for many non-linear probabilistic distance measures. Rather we propose a strategy, that allows us to construct spaces that allow the indirect estimation of the distances in Table 3.1
Table 3.1. Probabilistic distance measures between two probability distributions along with their expressions in terms of inner products (last column) between transformed versions of the probability functions. The transformations are represented by $Q_j = f_j(P_j)$ and $Q_{N+j} = f_{2j}(P_j)$ where $N$ represents the number of training relational distributions used to construct the configuration space.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chernoff distance</td>
<td>$\sum_j f_j(P_j) \log \left( \frac{f_j(P_j)}{P_j} \right) - \sum_j \log \left( f_j(P_j) \right)$</td>
</tr>
<tr>
<td>Bhattacharya distance</td>
<td>$\sum_j \left( f_j(P_j) - P_j \right)^2$</td>
</tr>
<tr>
<td>Matusita distance</td>
<td>$\sum_j \log \left( f_j(P_j) \right)$</td>
</tr>
<tr>
<td>KL divergence</td>
<td>$\sum_j f_j(P_j) \log \frac{f_j(P_j)}{P_j} - \sum_j f_j(P_j) \log f_j(P_j)$</td>
</tr>
<tr>
<td>Symmetric KL</td>
<td>$\sum_j f_j(P_j) \log \frac{f_j(P_j)}{P_j}$</td>
</tr>
</tbody>
</table>

19
We observe that the distance measures, such as the ones listed in Table 3.1, can be expressed as functionals of the inner product of transformed versions of the given probability functions. For instance, the Symmetric Kullback-Liebler (Symmetric KL) distance measure between two probabilities $P_1(a)$ and $P_2(a)$, given by $\sum_a (P_1(a) - P_2(a)) \log \frac{P_1(a)}{P_2(a)}$, can be expressed in terms of inner products as: $\langle f_1(P_1), f_2(P_1) \rangle - \langle f_1(P_1), f_2(P_2) \rangle - \langle f_1(P_2), f_2(P_1) \rangle + \langle f_1(P_2), f_2(P_2) \rangle$, where $f_1$ and $f_2$ represent the identity and the log transformations respectively, and $\langle a, b \rangle$ is the inner product (dot product for vectors) of $a$ and $b$. The third and fourth columns in Table 3.1 list the required transformation functions, denoted here by $f_1$ and $f_2$, for five different distance measures.
Let the given training set of $N$ relational distributions (or probability functions) that are used to construct the configuration space be denoted by \( \{ P_j(a) | j = 1, \cdots, N \} \). Let \( \{ Q_1, \cdots, Q_{2N} \} \) represent the set of $2N$ transformed probability functions, where the first $N$ transformed functions are given by \( \{ Q_j(a) = f_1(P_j(a)) | j = 1, \cdots, N \} \) and the next half is given by \( \{ Q_{j+N}(a) = f_2(P_j(a)) | j = 1, \cdots, N \} \). Note that we can now express the distance measures in terms of the $Q_j$ functions. For example, the Symmetric KL distance between two relational distributions $P_1$ and $P_2$, can now be written as $\langle Q_1, Q_{N+1} \rangle - \langle Q_1, Q_{N+2} \rangle - \langle Q_2, Q_{N+1} \rangle + \langle Q_2, Q_{N+2} \rangle$. The last two columns in the table describes how each distance can be computed using inner products of transformations of the probability functions. So, if we had a space that would allow us to compute these inner products of the transformed versions of the probabilities, i.e. inner product between the $Q_j$s, then we can construct the corresponding distance measure from them.

Let the entries of the $2N \times 2N$ inner product matrix, $S$, of these functions is given by

$$ s_{ij} = \langle Q_i(a) Q_j(a) \rangle = \sum_a Q_i(a) Q_j(a) \tag{3.1} $$

where $i, j = 1, \cdots, 2N$. We seek to find coordinates, $x_j$ for each of the $Q_j$'s, such that the inner product computed based on the coordinates match the inner products computed from the $Q_j$ functions, i.e. $x_i^T x_j = s_{ij}$. Expressing this in matrix notation, we need to find a matrix $X$, whose columns are the coordinates, such that

$$ S = X^T X \tag{3.2} $$

This we find by performing a Singular Value Decomposition (SVD) of the square matrix $S$ and choosing $X$ to be the eigenvectors, $U$, scaled by the square root of the eigenvalues, $\Lambda$.

$$ S = U \Lambda U^T = X^T X \tag{3.3} $$
The coordinates are given by

\[ X = \sqrt{\Lambda} U^T \]  \hspace{1cm} (3.4)

The dimension of embedding space can be determined by considering the eigenvalues associated with each dimension. We can retain the number of dimensions, \( p \), to keep a certain percentage of the energy, typically 99.5\%. The solution given by Eq. 3.3 and 3.4 is least squares in nature. Hence retaining higher number of dimensions results in a better approximation of the new coordinates in the embedding space.

It may be noted that the dot product matrix, \( S \), is a symmetric matrix and as we show next here, it is also a positive definite matrix. Let \( S = R^T R \) where \( R = [Q_1, \ldots, Q_{2N}] \) with each of \( Q_j \)'s being one column. For any nonzero vector \( y \), \( (Ry)^T (Ry) > 0 \), i.e. \( y^T S y > 0 \), and hence \( S \) is a positive definite matrix [101]. So, the eigenvalues are always greater than zero. As an illustration to the distance-based embedding, Fig. 3.2(a) shows the embedding of three 1D probability functions for the symmetric KL distance measure. Fig. 3.2(b) shows the embedding of a new probability function onto the low dimensional space, which we discuss next.

For a particular distance measure, if greater number of dimensions are retained, i.e. \( p \) is increased, then the final recognition accuracy increases. But different distance measures have different values of \( p \) for the same percentage of energy retained. It depends on the nature of the distance measure. For example, with Symmetric KL and KL, \( f_1 \) and \( f_2 \) transformations result in two very far off sets of points in \( \mathbb{R}^n \), and hence the same percentage of energy is retained in a fewer number of dimensions than that in Bhattacharya, Matusita or Chernoff distance where the two sets are either the same or closer to each other. Hence in order to achieve similar discrimination power, we need to keep a higher percentage of energy for the Symmetric KL and KL measures than the other three measures. We later
present results for various distance measures by varying the percentage of energy retained from 99.1% to 99.9% (Fig. 3.8).

3.3 Embedding New Samples

Typically the configuration space is constructed only once for the model (training) set of sequences, during the “training” phase. At runtime, relational distributions of the new sequences are embedded in this configuration space to compute distances to models. However, it may be noted that unlike other dimensionality reduction mechanisms such as principal component analysis, the dimensions of the configuration space do not have explicit representations. This means that we cannot embed new points into this space by projecting them onto the axes. In order to find the coordinates of a new point, we have to rely on distances of the new point from the previously embedded points. This is not a new problem. Such embedding of new points are known as “out of sample” embedding. However, our particular instance of the problem is novel. For every new relational distribution, $P_z(a)$, we have to embed two points, $f_1(P_z(a))$ and $f_2(P_z(a))$, such that distances of these two points from the currently embedded points are preserved along with the constraint of the distance between them. This implies that we cannot treat these two embeddings independently. We outline a coupled, iterative solution for this process.

Let $x_i$ denote the coordinate of the $i$-th embedded point, $Q_i(a)$. Recall that each $Q_i(a)$ is either $f_1(P_j(a))$ or $f_2(P_j(a))$, the $f_1$ and $f_2$ transformed versions of some relational distribution $P_j(a)$. For any new relational distribution, we would like to find the coordinates, $z_1$ and $z_2$ for $f_1(P_z(a))$ and $f_2(P_z(a))$ respectively, such that the inner products computed based on the embedded coordinates match the inner products computed from the raw func-
tions. In other words,

\[
\begin{bmatrix}
  x_1(1) & \cdots & x_1(p) \\
  \vdots & \vdots & \vdots \\
  x_{Na}(1) & \cdots & x_{Na}(p)
\end{bmatrix}
\begin{bmatrix}
  z_1(1) \\
  \vdots \\
  z_1(p)
\end{bmatrix}
= \begin{bmatrix}
  \langle f_1(P_z(a)), Q_1(a) \rangle \\
  \vdots \\
  \langle f_1(P_z(a)), Q_{Na}(a) \rangle
\end{bmatrix}
\]

(3.5)

and

\[
\begin{bmatrix}
  x_1(1) & \cdots & x_1(p) \\
  \vdots & \vdots & \vdots \\
  x_{Na}(1) & \cdots & x_{Na}(p)
\end{bmatrix}
\begin{bmatrix}
  z_2(1) \\
  \vdots \\
  z_2(p)
\end{bmatrix}
= \begin{bmatrix}
  \langle f_2(P_z(a)), Q_1(a) \rangle \\
  \vdots \\
  \langle f_2(P_z(a)), Q_{Na}(a) \rangle
\end{bmatrix}
\]

(3.6)

and

\[
z_1^T z_2 = \langle f_1(P_z(a)), f_2(P_z(a)) \rangle
\]

(3.7)

Eq. 3.5 and 3.6 represent the distance constraint to \( Na \) of the already embedded points. We will refer these \( Na \) points as the reference points. Eq. 3.7 is the constraint related to two newly embedded points. These equations can be expressed in matrix form as:

\[
Az_1 = c_1
\]
\[
Az_2 = c_2
\]
\[
z_2^T z_1 = c_{12}
\]

(3.8)

The optimal choice of the embedding points can then be expressed as

\[
\{z_1, z_2\}_{opt} = \arg\min_{z_1, z_2} E(z_1, z_2)
\]

(3.9)

This minimization has to be solved using an iterative alternating method [102]. Using the fact that

\[
\min_{z_1, z_2} E(z_1, z_2) = \min_{z_2} \min_{z_1} E(z_1, z_2)
\]

(3.10)
we can construct an iterative scheme for solving the above minimization, where at each step we first minimize with respect to \( z_1 \) and then with respect to \( z_2 \). Thus,

\[
\begin{align*}
  z_1^{t+1} &= \arg\min_{z_1} E(z_1, z_2') \\
  z_2^{t+1} &= \arg\min_{z_2} E(z_1^{t+1}, z_2)
\end{align*}
\]  

(3.11)

where we used the superscript \( t \) to denote the \( t \)-th iteration. Since \( E(z_1^{t+1}, z_2^{t+1}) \leq E(z_1^t, z_2^t) \) and \( E(z_1, z_2) \geq 0 \), i.e. bounded from below, we will produce a monotonically non-increasing set of estimates. Given the local nature of this alternating mechanism, the converging point could either be a minimum or a saddle point. To solve each of the alternating minimizations, \( \min_{z_1} E(z_1, z_2) \) and \( \min_{z_2} E(z_1^{t+1}, z_2) \), we use the fact that at the minimum point the derivative, \( \frac{\partial}{\partial z_1} E(z_1, z_2) \) and \( \frac{\partial}{\partial z_2} E(z_1^{t+1}, z_2) \), have to be zero. These conditions translate to

\[
\begin{align*}
  \frac{\partial}{\partial z_1} E(z_1, z_2') &= \frac{\partial}{\partial z_1} \left( (Az_1 - c_1)^T (Az_1 - c_1) + (z_2^T z_1 - c_{12})^2 \right) \\
  &= \frac{\partial}{\partial z_1} \left( (z_1^T A^T A z_1 - 2c_1^T A z_1 + c_1^T c_1) + (z_1^T z_2 z_2^T z_1 - 2z_2^T z_1 c_{12} + c_{12}^2) \right) \\
  &= 2z_1^T A^T A - 2c_1^T A + 2z_1^T z_2 z_2^T - 2z_2^T c_{12} \\
  &= 0
\end{align*}
\]

(3.12)

This condition can be rewritten in a compact form by augmenting \( A \) with \( z_2 \) as \( B_2^T = [A^T, z_2] \) and using \( d_1^T = [c_1^T, c_{12}] \).

\[
B_2^T B_2 z_1 = B_2^T d_1
\]

(3.13)

or

\[
z_1 = \left( B_2^T B_2 \right)^{-1} B_2^T d_1 = B_2^+ d_1
\]

(3.14)
where $B_2^\dagger$ is the pseudo-inverse of $B_2$. Equivalently, by considering the derivative with respect to $z_2$ we have
\[ z_2 = \left( B_1^T B_1 \right)^{-1} B_1^T d_2 = B_1^\dagger d_2 \] (3.15)
where $B_1$ is $A$ augmented with $z_1$ and $d_2$ is $c_2$ augmented with $c_{12}$. Eqs. 3.14 and 3.15 form the coupled equations that we solve at each iteration. The iterative equation solving process is stopped when the $L_2$ norms $|z_t^{t+1} - z_t^t|_2$ and $|z_t^{t+1} - z_t^t|_2$ are less than a given threshold. We used a threshold of 0.001 in all our experiments. We have found that, in practice, this process converges after a few iterations.

### 3.4 Choice of Reference Points

The choice of the number of reference points, $N_a$, is dependent on the dimension of the low-dimensional space $p$. In general, $N_a > p$. Later we experiment with this number. The choice appears to be dependent on the probabilistic distance measure. We choose the reference points by uniform random sampling with replacement from the set of transformed probabilities. For Chernoff, Symmetric KL and KL distances where $f_1$ and $f_2$ are the different functions, we sample the $N_a$ points from the set $Q_1, Q_2, \cdots, Q_{2N}$. On the other hand, for Bhattacharya and Matusita distances, where $f_1$ and $f_2$ are the same functions and the sets $Q_1, \cdots, Q_N$ and $Q_{N+1}, \cdots, Q_{2N}$ are same, we sample the $N_a$ reference points from the set $Q_1, Q_2, \cdots, Q_N$.

We have experimented with other strategies for choosing the reference points, such as choosing the $N_a$ subset of points whose inter-point distance is above a chosen threshold, or construct a subset of $N_a$ independent set of points in a greedy fashion. However, those methods did not give a consistently stable solution. Random selection seemed to work better. To show the stability of the random selection solution, we will show our results by repeating the experiments 25 times, each with different random set of reference points.
As we see later in Fig 3.7 in the results section, the accuracy increases and the standard deviation decreases with increase in the number of reference points for each of the shown distance measures. Depending on the nature of the probabilistic distance, the increase is seen more for some distance measures than others.

### 3.5 Time Complexity

We analyze the time complexity of finding the distance of a new sample from the existing $N$ samples with and without embedding. Let the dimension of each relational distribution before embedding be $M$. Then the time complexity of comparing a new relational distribution to all existing $N$ relational distributions is $O(NM)$, without performing embedding. On the other hand, with embedding, the time is $O((M + N)p + p^3)$, where $p$ is the number of dimensions retained in the low dimensional space. It includes the time required for embedding the new point into the low dimensional space by forming the set of linear equations ($O(Mp)$) and solving it ($O(p^3)$), and the time to compute distances from all other points in the $p$-dimensional space ($O(Np)$). Note that $p << M$ and as $N$ increases the advantage of having a low-dimensional space increases.

Each motion sequence is represented by two trajectories, corresponding to $f_1$ and $f_2$ in the configuration space. For distances like Matusita and Bhattacharya, where $f_1$ and $f_2$ are same, each sequence is represented by a single trajectory. The length of each trajectory is equal to the number of frames in the motion sequence. We match a test sequence with a training sequence using dynamic time warping.

### 3.6 Matching

Let $l_1$ and $l_2$ represent the length of the two sequences and $d(i, j)$ represent the probabilistic distance between the $i^{th}$ relational distribution from the first sequence and the $j^{th}$
relational distribution from the second sequence. \(d(i, j)\) is computed in terms of dot products of the coordinates of the transformed relational distributions embedded in the configuration space. Analogous expressions, as given in the Table 3.1 are used for various distance measures. Let \(D\) represent the score matrix \(D\) of size \((l_1 + 1) \times (l_2 + 1)\). The 0th row and column of \(D\) are initialized to infinity, except \(D(0,0)\), which is initialized to 0. The rest of the score matrix \(D\) is completed using the following recursion:

\[
D(i, j) = d(i, j) + \min \{D(i - 1, j), D(i - 1, j - 1), D(i, j - 1)\} \tag{3.16}
\]

where \(1 \leq i \leq l_1\) and \(1 \leq j \leq l_2\). \(D(l_1, l_2)\) then indicates the dissimilarity score between the two sequences. The lower \(D(l_1, l_2)\) is, the more similar the two sequences are. Each test sequence is matched to every training sequence and is finally labeled as belonging to the class of the training sequence with lowest dissimilarity score.

3.7 Results and Analyses

We illustrate the versatility of the configuration space ideas to represent and recognize motion patterns using (i) signs from American Sign Language (ASL), (ii) two-handed gestures from human-computer interaction domain, and (iii) classification of human-human interaction sequences. Using these datasets, we thoroughly analyze the different aspects of the representation. Specifically, we considered the following questions.

1. What are the recognition rates? What is the loss in accuracy due to dimensionality reduction? What are the time savings? How are the recognition rates across subjects?

2. What is the impact of the different probabilistic measures on recognition? Is using the Euclidean distances between distributions sufficient?
3. What is the impact of varying the number of reference points used for embedding distributions?

4. What are the trade-offs between the extent of dimensionality reduction and accuracy?

For all these studies, as a performance metric, we uniformly used the overall recognition rates of the signs or gestures or actions, as the case maybe. We adopted a task-based evaluation philosophy and decided against the use of any measure that evaluates the quality of any intermediate level representation such as the relational distribution. The accuracy reported throughout this section is the average of the accuracies obtained over 25 test runs with the reference points for the embedding process, sampled independently in each run based on uniform random sampling with replacement, from the set of $Q_1, \ldots, Q_{2N}$. Along with the average we report the standard deviation. The time reported is also averaged over the 25 runs.

3.7.1 Datasets

We start by describing the datasets, the train-test splits for each, and lower level primitives used in each case for constructing the relational distributions.

3.7.1.1 Signs from American Sign Language

This dataset comprises of 147 different ASL signs and is unique in the large number of classes involved. Fig. 3.3 shows five of them. The length of the signs varies between 2 to 33 frames, the average being approximately 10 frames and the resolution of each frame is 490x370 pixels. There are 2 instances of each sign, one is used for training and the other for testing. We choose Canny edges as our primitives. Skin color blobs are first extracted from the images. We use a Bayesian classifier with a histogram technique [103] for skin detection. Each pixel in a new image is classified as a skin pixel if its probability in the skin
histogram is more than that in the background histogram. The skin blobs’ contour pixels along with the Canny edges found inside them are selected as the low-level primitives. One might use the contour pixels alone, but in that case, the hand shape would be missed out completely when the hands come in front of the face.

3.7.1.2 Two-Handed Gestures

We used a subset of the two-handed gesture described in [104] that involves the variations of the same gesture across persons. There are 7 different kinds of gestures performed by 7 different persons. The resolution of the images is $320 \times 240$ pixels. Fig. 3.4 shows the 7 types of gestures. They are push, rotate-front, rotate-back, rotate-up, rotate-down, rotate-right, and rotate-left.

We will show two kinds of experiments with this dataset. For one, we used one session of each gesture performed by each person for training and two different sessions of each gesture by each person for testing. For the other, which is an across-person recognition experiment, we used 2 sessions of each of 7 types of gestures performed by 4 persons for training, and 10 sessions of each of the 7 types of gestures performed by 3 other persons for testing.

To generate the low-level features, we segment the hand blobs using color information. Then, we use contour pixels of the two hands as the low level primitives.

3.7.1.3 Human-Human Interaction

The third dataset we use is the Human Interaction data from CMU Graphics Lab Motion Capture (MoCap) Database [105]. We do not use the marker data, rather we use the video data of the persons performing the interactions. The 8 different classes of human interactions that we used in our experiments are ‘A pulls B, B resists’, ‘A sits, B pulls A’, ‘Chicken Dance’, ‘Walk together’, ‘Walk away from each other’, ‘Walk towards each
other’, ‘One gets up after a scramble for last seat’, and ‘One saves other’. Fig. 3.5 shows examples of the sequences. Each class of interaction has two instances. We used one for training and the other for classification. The resolution of the frames is 352x240 pixels. Blobs representing human bodies are extracted based on color information and the blob contour pixels are used as primitives. This dataset is particularly challenging because it is difficult to segment out the persons cleanly from the background as their dark clothings matches the darkness of some part of the background; segmentation is not perfect. This is depicted in the edge sequences shown in Figure 3.6. Note that the contours do not represent perfect body contours and they are effected by the objects in the background.

3.7.2 Recognition Performance

In the first set of experiments, we study the effectiveness of the representation in the diverse set of tasks, i.e. sign, gesture, and action recognition. We use fixed size relational distributions (51 x 51) in all our experiments. For each of the three datasets, we use the five different distance measures mentioned in Table 3.1. For the Chernoff distance, we used the values $\alpha_1 = \frac{1}{3}$ and $\alpha_2 = \frac{2}{3}$. In addition, we use the PCA on the relational distributions, followed by a Euclidean distance metric as another possible distance computation method. Depending on the probabilistic measure, each training sequence is represented by either a single sequence of coordinates (for Matusita and Bhattacharya) or two sequences of coordinates (for Chernoff, KL and Symmetric KL) in the low dimensional space. A test sequence similarly results in either one or two sequences of coordinates, obtained by embedding the transformations of relational distributions corresponding to frames in the test sequence onto the low dimensional space. This embedding process needs the choice of a set of embedding points to which we compute the distances. To generate recognition performance statistics, i.e. mean and standard deviation accuracy, we consider 25 possible random choices of this set for each test sequence. We match the test and each train embedding using dynamic time

31
Table 3.2. Recognition performance on the ASL dataset with 147 different test signs.

<table>
<thead>
<tr>
<th>Distance Measure</th>
<th>Using dimensionality reduction</th>
<th>Without dimensionality reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy% (Std. Dev.)</td>
<td>Time (in sec)</td>
</tr>
<tr>
<td>Matusita</td>
<td>74.6 (0.5)</td>
<td>0.97</td>
</tr>
<tr>
<td>Bhattacharya</td>
<td>80.3 (0.7)</td>
<td>0.61</td>
</tr>
<tr>
<td>Chernoff</td>
<td>75.5 (1.1)</td>
<td>1.05</td>
</tr>
<tr>
<td>Symmetric KL</td>
<td>77.2 (1.4)</td>
<td>2.33</td>
</tr>
<tr>
<td>KL</td>
<td>54.9 (3.9)</td>
<td>1.98</td>
</tr>
<tr>
<td>Euclidean(PCA)</td>
<td>76.2</td>
<td>1.11</td>
</tr>
</tbody>
</table>

Table 3.3. Recognition performance on the two-handed test gesture sequences based on 98 test sequences.

<table>
<thead>
<tr>
<th>Distance Measure</th>
<th>Using dimensionality reduction</th>
<th>Without dimensionality reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy% (Std. Dev.)</td>
<td>Time (in sec)</td>
</tr>
<tr>
<td>Matusita</td>
<td>98 (1.03)</td>
<td>3.11</td>
</tr>
<tr>
<td>Bhattacharya</td>
<td>95 (0.79)</td>
<td>1.78</td>
</tr>
<tr>
<td>Chernoff</td>
<td>96 (0.66)</td>
<td>2.88</td>
</tr>
<tr>
<td>Symmetric KL</td>
<td>94 (1.27)</td>
<td>3.29</td>
</tr>
<tr>
<td>KL</td>
<td>64 (9.34)</td>
<td>3.04</td>
</tr>
<tr>
<td>Euclidean(PCA)</td>
<td>96</td>
<td>3.61</td>
</tr>
</tbody>
</table>

warping based on the distances computed in the low-dimensional space and classify using the nearest neighbor classifier.

Tables 3.2, 3.3 and 3.4 list the recognition performances on the three datasets, for various distances measures, and with and without the low-dimensional embedding. We can make several observations.
Table 3.4. Recognition performance on the human interaction dataset for 8 test sequences.

<table>
<thead>
<tr>
<th>Distance Measure</th>
<th>Using dimensionality reduction</th>
<th>Without dimensionality reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># correct</td>
<td>Time (in sec)</td>
</tr>
<tr>
<td>Matusita</td>
<td>6 (0)</td>
<td>8.42</td>
</tr>
<tr>
<td>Bhattacharya</td>
<td>6 (0)</td>
<td>5.45</td>
</tr>
<tr>
<td>Chernoff</td>
<td>7.5 (0.5)</td>
<td>8.55</td>
</tr>
<tr>
<td>Symmetric KL</td>
<td>6.6 (0.5)</td>
<td>7.67</td>
</tr>
<tr>
<td>KL</td>
<td>4.8 (0.5)</td>
<td>7.18</td>
</tr>
<tr>
<td>Euclidean(PCA)</td>
<td>6</td>
<td>8.78</td>
</tr>
</tbody>
</table>

1. The drop in recognition rates with using a low-dimensional embedding is small. This is evident when we compare recognition rates with and without any dimensionality reduction.

2. There is about 2 to 3 times speed up in performance with the low-dimensional embedding. The time reported in the table and at other places is the time required to embed all relational distributions of a test sequence into the configuration space and to recognize the test trajectory by matching it to all the training trajectories using dynamic time warping. The time is averaged over all the test sequences. All the reported time indicate the CPU time on a 3 GHz Xeon workstation with 2 GB of memory.

3. The representation also holds promise for recognition across persons. From the two-handed gesture dataset, we used 2 sessions of each of 7 types of gestures performed by 4 persons for training, and 10 sessions of each of the 7 types of gestures performed by 3 other persons for testing. The persons used for testing were not used for training. Altogether, we used 56 training sequences and 210 test sequences. We obtained an average recognition accuracy of 73% with a standard deviation of 1.1% across the 25
test runs. It is based on the Matusita distance measure, which seems to be the best measure to use of this dataset.

4. The choice of the probabilistic distance measure matters. For the ASL dataset, the Bhattacharya distance gives the best result, while for the two-handed gestures dataset, Matusita distance measure gives the best result. The KL measure results in the poorest performance, which can be due to its inherent asymmetric nature of distance between the two probabilities. The performance of the PCA+Euclidean approach is not consistently high.

The differences among the performances with different distance measures are accentuated when trying to match a low temporal resolution sequence to a high temporal resolution one. For the two-handed gesture dataset, we created test sequences that are temporally sub-sampled from the original resolution; we retained one out of every five frames. The train sequences remained at original resolution. The recognition accuracies were 61\%, 69\%, 66\%, 67\% and 59\% for the Matusita, Bhattacharya, Chernoff, Symmetric KL, and PCA+Euclidean, respectively. This shows the recognition accuracies spread over a wider range and some of the probabilistic distances outperform the PCA+Euclidean approach by large margins.

We also experimented by considering only the odd frames of the training sequences and even frames of the test sequences, using the Matusita distance measure. The number of dimensions and the number of reference points used were same as that of the experiment in Table 3.3 for the Matusita measure. The average accuracy obtained over 25 test runs is 94\%. This shows the robustness of our approach.
3.7.3 Choice of Number of Reference Points

How do the choice of the number of reference points for embedding effect the quality of recognition? Fig 3.7 shows the results of an experiment designed towards this question. For the two-handed gesture dataset, we repeated the recognition experiments with $2p$, $3p$, $\cdots$, $10p$ number of reference points, where $p$ is the dimension of the low-dimensional embedding. We see from the plots that for Matusita, Bhattacharya, and Chernoff the effect on accuracy is quite small. For Symmetric KL, the accuracy increases initially with the increase in the number of reference points and then saturates. An analogous trend is seen with the standard deviation of the accuracy. As expected, the time taken to match the sequences increases with the increase in the number of reference points. However, it should be noted that the time required with even $10p$ reference points is less than the time taken without dimensionality reduction for each of the distance measures (the last column in Table 3.3).

3.7.4 Effect of the Dimensions of the Embedding

With the increase in percentage of energy retained during the dimensionality reduction, the number of dimensions and accuracy varies differently depending on the nature of the distance measure. Fig. 3.8 shows the variations for different probabilistic measures for the two-handed gesture dataset. As it can be seen, a higher percentage of energy is required for Symmetric KL measure than the other three types of measures to obtain the same number of dimensions or the same accuracy. The reason for this is the presence of the log transform in the Symmetric KL measure.
Figure 3.2. Embedding probability density functions. (a) Three 1D probability density functions, \(a\), \(b\) and \(c\), each being 50 dimensional, are embedded into a 6-dimensional space. The Symmetric KL distance between them, both before and after the dimensionality reduction are shown as images. It should be noted that all the pairwise distances are preserved in the embedding. (b) shows a new probability density function, \(d\). Two transformed functions of \(d\), \(f_1(d)\) and \(f_2(d)\) are computed, which are embedded onto the 6-dimensional space. It should be noted that the distance of \(d\) from the existing functions \(a\), \(b\), \(c\) in the low dimensional space are very close to those in the original space before embedding.
Figure 3.3. Five ASL signs (best viewed in color). The complete test dataset consists of 147 different signs. For display purposes, intermediate frames have been skipped for some of the signs shown above and the displayed ones are reduced in size.
Figure 3.4. Seven different types of two-handed gestures (best viewed in color). For display purposes, intermediate frames have been skipped and the displayed ones are reduced in size. (Source: IDIAP Research Institute, Switzerland).
Figure 3.5. Eight different types of human interaction sequences (best viewed in color). For display purposes, intermediate frames have been skipped and the displayed ones are reduced in size. (Source: Carnegie Mellon University Graphics Lab).
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) A Pulls B, B Resists</td>
<td>(b) A sits, B pulls A</td>
<td>(c) Chicken Dance</td>
<td>(d) Walk together</td>
<td>(e) Walk away from each other</td>
<td>(f) Walk towards each other</td>
<td>(g) One gets up after a scramble for last seat</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(h) One saves other</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.6. Extracted contours for the sequences in Figure 3.5. It should be noted that due to the presence of similar colored objects in the background the blobs are not perfectly segmented.
Figure 3.7. Effect of the number of reference points used during embedding. (a) shows the average accuracy obtained over 25 test runs, (b) shows the standard deviation of the accuracy, and (c) shows the average time taken to embed and recognize a given series of relational distributions.
Figure 3.8. Effect of the percentage of energy retained with various distance measures. (a) shows the average accuracy obtained over 25 test runs, (b) shows the number of dimensions for each type of distance measure for a range of the percentage of energy retained.
4.1 Time for Building the Relational Distributions

Relational distributions, incorporating the shape aspect of an object in an image sequence, have been used earlier in the representation of periodic human motion [21] and continuous sign language sentences [29, 106]. However, the naive approach to compute relational distributions would involve an exhaustive enumeration of all pairs of edge pixels which is computationally intensive with a time complexity of \( O(n^2) \), where \( n \) is the number of edge pixels in an image. For instance for the ASL dataset, it takes 0.665 seconds to compute a \( 51 \times 51 \) relational distribution for one image. This has been the practice in earlier uses of relational distributions.

We have found that using a sampling-based method to estimate the relational distribution offers an efficient alternative. At each iteration, we sample \( m \) pairs of edge pixels with replacement, where \( m \) is directly proportional to the number of histogram bins. We repeat until the change in entropy of the distribution is small. The average complexity of computation of a relational distribution is thus reduced to \( O(km) \), where \( k \) is the number of iterations for which the relational distribution is updated. From practice we have found that the number of iterations required for a relational distribution to converge, \( k \), and the number of pairs of edge samples sampled at each iteration, i.e. \( m \), are smaller than \( n \). Fig. 4.1 shows an example of the quality of relational distributions obtained in this manner. Fig. 4.1(b) and (c) show the relational distribution computed by exhaustive sampling and random sampling.
Figure 4.1. Example of relational distributions constructed for the edge image in (a) using (b) exhaustive sampling and (c) random sampling. (d) shows the change in entropy with random sampling iterations, respectively for the edge image in (a). Fig. 4.1(d) shows the change in entropy with each iteration.

Obviously, there is some fall in the fidelity of representation with sampling. Does this loss impact final recognition rates? For this, we studied the impact on the recognition rates on the ASL data set and the time to estimate the relational distribution with number of sampling iterations. We varied $m$ from 1% to 5% of the total number of bins in the histogram ($51 \times 51 = 2601$ in our experiments), and $k$ was varied from 5 to 30 iterations at an interval of 5 iterations. Using exhaustive sampling, the average time taken to compute a
Figure 4.2. Effect of using sampling methods to estimate the relational distributions. For different percentages of total number of possible samples, (a) shows the overall accuracy obtained with the ASL signs dataset using the Bhattacharya measure and (b) shows the average time taken to estimate one relational distribution.

A relational distribution for each image in the test dataset was 0.665 seconds with an overall accuracy of 79% using the Bhattacharya distance measure. Fig. 4.2 shows the results based on the random sampling approach. As can be seen, the time to compute a single relational distribution is much less with the random sampling approach while there is a very little fall in accuracy. For example, using a sampling set size of 2% of the number of bins in a histogram, and improving the histogram over 30 iterations of sampling, takes around one-third of the time taken with exhaustive sampling. The reduction in accuracy, from 79% to 77%, is much less compared to the computational savings achieved. We used the same set of randomly sampled 2p reference points for the above comparisons.

Another possible way to sample could have been to sample an edge image across scale space. Starting from a higher scale edge map, iterate to a finer scales. The idea is that gross structure would be captured at high scales and finer levels would be progressively captured until there no other significant structure to add. However, we have found our simpler sampling idea was sufficient for our experiments. For storage purposes, it should be noted that
Figure 4.3. Illustration of robustness of relational distributions with respect to parameters. (a) Edges extracted with different parameters in a frame from a sequence. (b) Bhattacharya distance of the relational distribution of each frame in a sequence from the first frame. Each plot corresponds to a different edge detection parameter.

the relational distributions are sparse matrices and all existing efficient implementations for sparse matrices could be used for them. This would result in further savings.

4.2 Robustness with Imperfect Edges

Here, we discuss how robust are the relational distributions to the edge detection parameters used to find the edges? This study of robustness of the shape descriptors is important for the contour-based shape recognition algorithms [107]. In Fig. 4.3 we show an illustration of the stability of the representation for a motion sequence representing one ASL sentence. We compute Bhattacharya distance between the relational distribution for each frame from that for the first frame in the sequence. We repeat this plot for each edge detection parameter, in this case it is the Canny edge high threshold, which controls the amount of “clutter” at any given scale. Different plots are obtained by varying the Canny edge high threshold value from \( \frac{50}{255} \) to \( \frac{200}{255} \) in intervals of \( \frac{5}{255} \). If the representation is stable the plots should cluster well and the variation between the curves at each frame should be lower.
than the variation of each plot over the sequence. The latter represents the variability of the configurations in the ASL sentence, which we expect to be high. We see from Fig. 4.3 that this is true, at least for this sequence.

We experiment with the impact of varying edge detection parameters on overall recognition rates. We use the Bhattacharya distance measure and ASL signs dataset (3.7.1.1) for this purpose. We use the Canny edge high threshold value of $\frac{50}{255}$ for all the train sequences and vary the threshold values for the test sequences. The recognition accuracies obtained using the same set of $2p$ reference points (3.4) were 76%, 77%, 80%, 78%, 67%, 64%, and 60% for the threshold values of $\frac{10}{255}$, $\frac{30}{255}$, $\frac{50}{255}$, $\frac{75}{255}$, $\frac{100}{255}$, $\frac{125}{255}$, and $\frac{150}{255}$ respectively. This shows a graceful degradation in accuracy as the difference between the threshold values used for train and test sequences is increased.

### 4.3 Interpolation

In many gesture-related applications, it is desirable to be able to match two gesture sequences performed at different speeds or captured at different frame rates. In some cases, the sampling rate might be very sparse or the frames of two motion sequences performed at the same speed might not be aligned – the sequences might have an offset between them. Dynamic time warping is a commonly used technique in such situations, but its results are more accurate if the two sequences being compared are at nearby temporal scales. In this section, we describe an approach to normalize the gesture sequences with respect to varying speeds by interpolating the series of points in the configuration space. For this, we perform cubic spline interpolation in the configuration space.

Interpolation between two configurations lets us arrive at a continuous representation of the motion in terms of change in the underlying structure. For further matching purposes, the motion is then indexed by arc length along the interpolated curve. The spline interpo-
lation is evaluated at equidistant points along the arc length. In our experiments, we used a fixed sampling distance that is equal to the mean of the $L_2$-norm distances between all consecutive points in all the motion sequences in the training dataset. Finer sampling can be done by using a fraction of the mean distance as the sampling distance. The resulting equidistant point series representation is the speed-normalized representation for a given motion sequence. Two nearby points along arc length of the interpolated curve, represent lesser amount of change in configuration than two points farther away along the curve.

Fig 4.4 depicts an example of the representation of motion as a smooth curve. Note that the variation in the rate of change in configuration between consecutive frames shows up in non-uniform spacing of the points (Fig 4.4(a)). This is overcome by interpolation (See Fig 4.4(b)). Further, the overall shape of the curves remains similar in both the interpolated curves and the number of sampled points is also nearly same in both of them. As we show below, in some cases, better recognition results can be obtained by using these speed normalized interpolated sequences as inputs to dynamic time warping, than using the raw non-interpolated points of motion trajectories. It should be noted that while interpolating along the arc length of a curve, we do not have any knowledge regarding the speed of the motion sequence. Each sequence (test or train) is first speed normalized separately by interpolating along arc length of its motion trajectory in the configuration space. A fixed sampling distance is used for all the sequences. The interpolated curves are then matched using dynamic time warping to find a dissimilarity score between a test and a train sequence, and the test sequence is labeled as the class of the train sequence with smallest dissimilarity score. If the temporal scale variation is known, then motion can also be interpolated in the spatio-temporal space along the temporal axis. We compare results of different kinds of interpolations.

An added advantage of a continuous curve representation in terms of cubic splines is the ability to compute derivatives, which in our configuration space would represent either rate
Figure 4.4. Example of interpolation in configuration space. (a) First three coordinates corresponding to image frames from two sequences, each representing a part of a motion sequence. Alternate frames in the sequence are dropped in one to simulate differences in speed/frame rate. The original image frames are shown with their corresponding points. (b) Resampled sequences after cubic spline interpolation along arc length of each curve without the knowledge of existing speed differences.
of change in configuration with time or amount of change in configuration along the arc. It should be noted that $L_2$-norm is closest in nature to the Matusita distance measure. Hence, we perform all the interpolation experiments using this distance measure. Interpolation based on distances like Chernoff, Symmetric KL and KL would need one to adopt a more involved strategy and adjust the sampling intervals to be faithful to original distances; we do not address that in this dissertation.

To demonstrate the effectiveness of the interpolation scheme based on the representation scheme, we used temporally sub-sampled test sequences of the two-handed gesture dataset. We kept one out of $k$ frames ($k = 1, 2, 3, 4, 5$) in the test sequences and matched them with the train sequence at original sequence. We performed the matching based on interpolation along arc length of the curves in the configuration space without the knowledge of the temporal scale variation, interpolation in the spatio-temporal configuration space knowing the temporal scale variation($k$), and with no interpolation. We used one set of randomly sampled $2p$ reference points.

Fig. 4.5(a) and (b) respectively show the overall recognition results and the time taken to recognize. As it can be observed, interpolation in the configuration space remarkably enhances the accuracy of recognition. It should also be noted that recognition with interpolation in the reduced dimensional space is faster than that in the original space, and there is no significant decrease in recognition accuracy.
Figure 4.5. Recognition results on the two-handed gesture dataset at various temporal scales (x to 5x). Experiments include spline interpolated data along the arc length without using any temporal scale information, spline interpolated data along temporal axis using temporal scale information, and non-interpolated data. (a) shows the recognition scores (b) shows the average time taken to recognize a sequence of relational distributions. (The legend in (a) applies to (b) as well.)
CHAPTER 5
UNSUPERVISED EXTRACTION OF COMMON MOTION PATTERNS FROM CONTINUOUS SENTENCES

Sign language sentences are series of signs with movement epenthesis in between signs. Figure 5.1 illustrates the traces of first vs. second dimensions in a low dimensional space of three sentences $S_1$, $S_2$ and $S_3$, with only one common sign, $R$, among them. Signeme represents the portion of the sign that would be present in all the sentences containing the sign, and is robust to the variations of adjacent signs. In this chapter, we present an unsupervised approach to automatically extract the signemes from multiple continuous sentences. Our approach can also be applied for extracting common words from spoken forms of the signed sentences. Since ASL grammar is not the same as that of spoken English, existing speech recognition softwares cannot be directly used for modeling the spoken words representing the signs in sentences. We model both the audio and video forms of the signs from continuous sentences (spoken and signed respectively). These models can be used in audio annotated video training frameworks where the modeler attaches audio annotations to training videos of ASL.

Fig 5.2 depicts the overview of our approach. Here, we present the theory in the context of signeme (video) without any loss of generality. The framework is also applicable for extracting common spoken words from sentences. We assume that the sign to be extracted is the only sign that is common to each sentence in the given set of $n$ sentences. Skin color blobs are extracted from frames of color video, and a relational distribution is formed for each frame using the edge pixels in the skin blobs. Each sentence is then represented as
Figure 5.1. Concept of signemes. First vs. second dimensions of sentences $S_1$ with signs $R_{11}, R, R_{12}$ in order, $S_2$ with signs $R_{21}, R, R_{22}$ and $S_3$ with signs $R_{31}, R, R_{32}$. The common sign is $R$. The portion of $R$ that is most similar across sentences is the signeme representative of $R$.

a trajectory in a low dimensional space called Space of Relational Distributions, which is arrived at by performing Principal Component Analysis on the relational distributions. The starting locations $(a_1, \ldots, a_n)$ and widths $(w_1, \ldots, w_n)$ of the candidate signemes in all the $n$ sentences are together represented by a parameter vector. It is initialized with random starting locations based on uniform random sampling from each sentence and initial width values are randomly selected from a given range of values. The parameter vector is updated sequentially by sampling the starting point and width of the possible signeme in each sentence from a joint conditional distribution that is based on the locations and widths of the target possible signeme in all other sentences. The process is iterated till the parameter values converge to a stable solution. Monte Carlo approaches [108, 109] like the Gibbs sampling [110], which is a special case of the Metropolis-Hastings algorithm [111] can be used for global optimization while updating the parameter vector by performing importance sampling on the conditional probability distribution. But it has a high burn-in period. In this dissertation, we adopt a greedy approach based on the use of Iterated Conditional Modes (ICM) [112]. ICM converges much faster than a Gibbs sampler, but is known to be largely dependent on the initialization. We overcome the limitation by performing ICM a number of times equal to the average length of the $n$ sentences, with different initialization.
tions. The most frequently occurring solution from all the ICM runs is considered as the final solution. We test our algorithm on both sign language videos and an audio dataset and show that our approach is robust to the variations in signs due to different contexts.

Optimize parameter vector \[\{a_1, w_1, \ldots, a_n, w_n\}\] using Iterated Conditional Modes (ICM).

Compute the following in a sequential manner:

\[
a_i^0, w_i^0 = \arg \max \left( f \left( a_1^0, w_1^0 / (a_1^1, w_1^1, \ldots, a_i^{i-1}, w_i^{i-1}) \right) \right)
\]

\[
a_i^1, w_i^1 = \arg \max \left( f \left( a_i^0, w_i^0 / (a_1^1, w_1^1, \ldots, a_i^{i-1}, w_i^{i-1}) \right) \right)
\]

and so on.

Repeat the above steps with \(i = (i + 1)\), till there is no change in values of the parameter set \(\{a_i, w_i\}\).

Select the initial parameters, \(\{a_1^0, w_1^0, \ldots, a_n^0, w_n^0\}\) based on uniform random sampling.

Perform PCA on the relational distributions to arrive at the Space of Relational Distributions.

Compute Relational Distribution for each frame capturing the pair-wise statistics of the low-level primitives.

Low level processing to extract primitives that most likely are from moving object.

Figure 5.2. Overview of our approach along with samples of intermediate representations. Each of the \(n\) sentences is represented as a sequence in the Space of Relational Distributions, and the common pattern is extracted using Iterated Conditional Modes (ICM). The parameter set \(a_1, w_1, \ldots, a_n, w_n\) is initialized using uniform random sampling and the conditional density corresponding to each sentence is updated in a sequential manner. \(\{a_1, w_1, \ldots, a_n, w_n\}\) denotes the parameter set at the end of \(i^{th}\) iteration of an ICM run. Multiple ICM runs are made, each with a different starting parameter vector and the most frequently occurring solution represents the set of common patterns, i.e. signemes.
5.1 Problem Formulation

Table 5.1. Notations used in this chapter

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{\vec{S}_1, \cdots, \vec{S}_n}</td>
<td>Set of (n) sentences with ONE common sign present in all the sentences. The index within a sentence could represent time or arc length in configuration shape space</td>
</tr>
<tr>
<td>(L_i)</td>
<td>Length of sentence (S_i)</td>
</tr>
<tr>
<td>(\vec{s}_{a_j}^{w_j})</td>
<td>Subsequence of sentence (S_j) starting from index (a_j) to (a_j + w_j - 1). We may sometimes use (\vec{s}_{j,a_j}^{w_j}) to make explicit the (j)-th index if it is not represented along with any other superscript or subscript of this term.</td>
</tr>
<tr>
<td>(A, B)</td>
<td>Possible choices of width for signemes of a sign include all integers from (A) to (B). The values of (A) and (B) are decided based on the dynamics involved in the sign.</td>
</tr>
<tr>
<td>(\theta)</td>
<td>Set of parameters ({a_1, w_1, \cdots, a_n, w_n}) defining a set of substrings of the given sentences</td>
</tr>
<tr>
<td>(\theta_{(a_i)})</td>
<td>Set of all parameters excluding the parameter (a_i). We have similar interpretations for (\theta_{(w_i)}) or (\theta_{(i)}).</td>
</tr>
<tr>
<td>(d(\vec{x}, \vec{y}))</td>
<td>Distance between the subsequences (\vec{x}) and (\vec{y}) based on a mapping found using dynamic time warping (DTW). This distance has to be calculated carefully so that it is not biased towards finding short subsequences only.</td>
</tr>
</tbody>
</table>

We formulate the signeme extraction problem in a probabilistic framework. Table 5.1 defines the notations that would be used in this chapter. We formulate the signeme extraction problem as finding the most recurring pattern among a set of \(n\) sentences \(\{\vec{S}_1, \cdots, \vec{S}_n\}\), that have one common sign present in all the sentences. The commonality concept underlying the definition of a signeme can be cast in terms of distances. Let \(\vec{s}_{a_i}^{w_i}\) represent a substring from the sequence \(\vec{S}_i\) consisting of the points with indices \(a_i, \cdots, a_i + w_i - 1\), and \(d(\vec{x}, \vec{y})\) denote the distance between two substrings \(\vec{x}\) and \(\vec{y}\) based on dynamic time warping. We define the set of signemes to be the set of substrings denoted by \(\{\vec{s}_{a_1}^{w_1}, \cdots, \vec{s}_{a_n}^{w_n}\}\) that is most likely among all possible substrings from the given set of sentences. Let \(\theta = \{a_1, w_1, \cdots, a_n, w_n\}\) denote the parameter set representing a set of substrings, and \(\theta_m\)
denote the parameter set representing the target set of signemes. Thus,

$$\theta_m = \arg \max_\theta p(\theta)$$  \hspace{1cm} (5.1)$$

where $p(\theta)$ is a probability over the space of all possible substrings. We define this probability to be a function of the inter-substring distances.

$$p(\theta) = \frac{g(\theta)}{\sum_\theta g(\theta)}$$  \hspace{1cm} (5.2)$$

where

$$g(\theta) = \exp \left( -\beta \sum_{i=1}^{n} \sum_{j=1}^{n} d(s_{ai}^{\mu}, s_{aj}^{\nu}) \right)$$  \hspace{1cm} (5.3)$$

$\beta$ is a positive constant. Note that $g(\theta)$ varies inversely with the summation of the pairwise distances of all the subsequences given by $\theta$. Also note that $p(\theta)$ is hard to compute or even sample from because it is computationally expensive to compute the denominator in Eq. 5.2 since it involves the summation over all possible parameter combinations. $\beta$ acts as a scale parameter, which controls the slopes of the peaks in the probability space. It can also be looked upon as the smoothing parameter. If probability sampling algorithms like Gibbs sampling [110] are used in later steps, then the rate of convergence would be determined by this parameter. While there is some freedom in choosing the function $g(\theta)$, we have to be careful that not all forms will give us the local form for the conditional density function that we discuss below. For instance, reciprocal of the total distance, i.e. $1/(\sum_{i=1}^{n} \sum_{j=1}^{n} d(s_{ai}^{\mu}, s_{aj}^{\nu}))$, is not particularly good in that respect.

To make sampling easier, we construct conditional density function of each of the parameters $\theta_i$, i.e. $\{a_i, w_i\}$, given the values of the rest of the parameters, i.e. $\theta_{(i)}$ or $\{a_1, w_1 \cdots a_{i-1}, w_{i-1}, a_{i+1}, w_{i+1} \cdots a_n, w_n\}$. In other words, we construct probability density function of the possible starting points and widths in each sentence, given the estimated starting
points and widths of the common pattern in all other sentences, i.e. \( f(\theta_i|\theta_{(i)}) \). Of course, this conditional density function has to be derived from the joint density function specified in Eq. 5.2.

\[
f(\theta_i|\theta_{(i)}) = \frac{p(\theta)}{p(\theta_{(i)})} = \frac{p(\theta)}{\sum_{\theta_i} p(\theta)} = \frac{g(\theta)}{\sum_{\theta_i} g(\theta)}
\]  

(5.4)

Since the normalization to arrive at this conditional density function involves summation over one parameter, it is now easier to compute and sample from. The specific form for this conditional density function using the dynamic time warping (DTW) distances is

\[
f(\theta_i|\theta_{(i)}) = \frac{\exp\left(-\beta \sum_{k=1}^{n} d(\vec{s}_{w_i}^{a_i}, \vec{s}_{w_k}^{a_k})\right)}{\sum_{\theta_i} \exp\left(-\beta \sum_{k=1}^{n} d(\vec{s}_{w_i}^{a_i}, \vec{s}_{w_k}^{a_k})\right)}
\]

(5.5)

where \( \theta_i \) represents the parameters from the \( i^{th} \) sentence, i.e. \( a_i \) and \( w_i \). Note that the distance terms that do not involve \( a_i \) and \( w_i \), i.e. do not involve the \( i \)-th sentence appear both in the numerator and the denominator and so cancel out. For notational convenience, this is sometimes represented using conditional \( g \) functions as:

\[
f(\theta_i|\theta_{(i)}) = \frac{g(\theta_i|\theta_{(i)})}{\sum_{\theta_i} g(\theta_i|\theta_{(i)})}
\]

(5.6)

where \( g(\theta_i|\theta_{(i)}) = \exp\left(-\beta \sum_{k=1}^{n} d(\vec{s}_{w_i}^{a_i}, \vec{s}_{w_k}^{a_k})\right) \).

5.1.1 Choice of Distance Measure

The distance function \( d \) in the above equations needs to be chosen carefully such that it is not biased towards the shorter subsequences. Here we briefly describe how we compute distance between two substrings using dynamic time warping.

Let \( l_1 \) and \( l_2 \) represent the length of the two substrings and \( e(i, j) \) represent the Euclidean distance between the \( i^{th} \) data point from the first substring and the \( j^{th} \) data point
from the second substring. Let $D$ represent the score matrix of size $(l_1 + 1) \times (l_2 + 1)$. The $0^{th}$ row and $0^{th}$ column of $D$ are initialized to infinity, except $D(0,0)$, which is initialized to 0. The rest of the score matrix, $D$, is completed using the following recursion:

$$D(i, j) = e(i, j) + \min \{D(i - 1, j), D(i - 1, j - 1), D(i, j - 1)\}$$

(5.7)

where $1 \leq i \leq l_1$ and $1 \leq j \leq l_2$. The optimal warp path is then traced back from $D(l_1, l_2)$ to $D(0,0)$. The distance measure between the two substrings is then given by $D(l_1, l_2)$ normalized by the length of the optimal warping path. We use this measure in our experiments.

One might as well experiment with other possible choices of the distance function like the sum of the mean Euclidean distances along the warped path with the maximum Euclidean distance along the warped path. But the former option sufficed for our experiments.

### 5.2 Parameter Estimation

In order to extract the common sign from a given set of sign language sentences, we need to compute $\theta_i$ for each of the sentences sequentially. Gibbs sampling [110] is a Markov Chain Monte Carlo approach [109] that allows us to sample the conditional probability density $f(\theta_i|\theta_{(i)})$ for all the sequences sequentially and then iterate the whole process till convergence. Gibbs sampling results in a global optimum, but its convergence is very slow. The burn-in period is typically thousands of iterations. So, we perform the optimization using Iterated Conditional Modes(ICM), first proposed by Besag [112]. ICM has much faster convergence, but it is also known to be heavily dependent on the initialization. We address this limitation by running the optimization multiple times with different initializations and choosing the most frequently occurring solution as the final solution.
Algorithm 5.2.1: Iterated Conditional Modes($\{a^0_1, w^0_1, \cdots, a^0_n, w^0_n\}$)

**comment:** Chooses $(a_1,w_1,\cdots,a_n,w_n)$ that maximizes the distribution $p(a_1,w_1,\cdots,a_n,w_n)$

**comment:** Initialization:

$\theta_0 \leftarrow \{a^0_1, w^0_1, \cdots, a^0_n, w^0_n\}$

repeat

for $i \leftarrow 0$ to $n$

**comment:** Jointly sample $a_i, w_i$. $L_i$ is the length of sequence $S_i$

for $w_i \leftarrow A$ to $B$

do

for $a_i \leftarrow 0$ to $L_i - w_i + 1$

do

$g(a_i, w_i|\theta(a_i, w_i)) \leftarrow \exp\left(-\beta \sum_{k=1}^n d(\vec{s}_a^{w_i}, \vec{s}_a^{w_k})\right)$

**comment:** Normalize

for $w_i \leftarrow A$ to $B$

do

for $a_i \leftarrow 0$ to $L_i - w_i + 1$

do

$f(a_i, w_i|\theta(a_i, w_i)) \leftarrow \frac{g(a_i, w_i|\theta(a_i, w_i))}{\sum_{a_i, w_i} g(a_i, w_i|\theta(a_i, w_i))}$

$a_i, w_i \leftarrow \text{ARG MAX } (f(a_i, w_i|\theta(a_i, w_i)))$

until CHANGE IN PARAMETERS($\{a_1, w_1, \cdots, a_n, w_n\}$) == 0

Algorithm 5.2.1 shows the use of ICM for extracting the common patterns or signemes from a set of sentences with a given initial parameter vector. We aim to select the set of parameters that maximizes the probability $p(\theta)$ or $p(a_1, w_1, \cdots, a_n, w_n)$. We do that by estimating each of the parameters $a_1, w_1, \cdots a_n, w_n$ in a sequential manner. Since we expect the starting location and width of a subsequence representing the common sign to be strongly correlated, we estimate $a_i$ and $w_i$ jointly. First we compute $g(\theta_i|\theta_{(i)})$ i.e.
\( g(a_i, w_i | \theta_{(a_i, w_i)}) \) from which we compute the conditional density functions \( f(\theta_i | \theta_{(i)}) \) i.e. \( f(a_i, w_i | \theta_{(a_i, w_i)}) \). Note that it involves a summation over \( a_i \) and \( w_i \) only, which involves much less computation than that required for computing \( p(\theta) \) which involves a summation over \( a_1, w_1, \cdots a_n, w_n \). The values for \( a_i \) and \( w_i \) are updated with those that maximize the conditional density \( f(\theta_i | \theta_{(i)}) \). The process is carried out sequentially for \( i = 1 \) to \( n \), and then repeated iteratively till the values of the parameter vector \( \{a_1, w_1, a_2, w_2, \cdots a_n, w_n\} \) do not change any more. Fig 5.4 shows an example of how the conditional probability \( f(\theta_{a_i, w_i} | \theta_{(a_i, w_i)}) \) changes for the first seven sentences from a given set of fourteen video sentences containing a common sign ‘DEPART’. The vertical axis in the probabilities represents the starting locations and the horizontal axis represents the possible widths. The brighter regions represent a higher probability value. Note that the probabilities are spread out in the first iteration for each sentence and it slowly converges to a fixed starting point for each of them. They remain more spread out across the horizontal(width) axis because we vary the width only in a small range of \( A \) to \( B \) for each sign, that is decided based on the amount of motion present in the sign. Fig. 5.3 plots the typical convergence of the parameter values in a single ICM run. It plots the norm of difference between consecutive parameter vectors versus the parameter vector update count, which is incremented each time a parameter is sampled or selected from the probability distribution \( f(\theta_i | \theta_{(i)}) \). It shows that ICM converges in less than \( 56/14 = 4 \) iterations. This, in turn, also indicates the local nature of the optimization achieved with ICM. The initialization is very important in this case. In the next subsection, we describe how we address this problem.

### 5.3 Sampling Starting Points for ICM

In order to address the local convergence nature of ICM, we adopt a uniform random sampling-based approach. We start by randomly assigning values to the parameter vector \( \theta \).
Figure 5.3. Convergence of values of the parameter set. The above plot shows the norm of the difference between two consecutive parameter vectors representing the set of starting points and widths of the common subsequence in the given set of sequences. It shows the typical convergence with a given initialization vector. ICM is repeated with multiple initializations and the most frequently occurring solution is considered as the final solution.

The width $w^0_i$ is obtained by sampling a width value based on uniform random distribution from the set of all possible widths ($A \cdots B$) for the sign. Then the value for $a^0_i$ is obtained by sampling a starting point based on uniform random distribution from the set of all possible starting points in the $i^{th}$ sequence, i.e. from the set \{1 \cdots (L_i - w^0_i + 1)\}. Different initial parameter vectors are obtained by independently sampling the sentences multiple times. ICM is run using each initial parameter vector generated and the most common solution is considered as the final solution. The uniform sampling of the frames in the sentences for selecting the starting locations ensures the whole parameter space is covered uniformly. The number of times we sample the initial parameter vector and run the ICM algorithm decides how densely we cover the whole parameter space. We run it the number of times equal to the average number of frames in each sentence from the given set of sentences for
Figure 5.4. Convergence of the conditional probability density $f(\theta_i|\theta(i))$ for sentences $S_1...S_6$ from a given set of sentences $S_1...S_{14}$. The brighter regions represent a higher probability value. The vertical axis in the probabilities represents the starting locations and the horizontal axis represents the possible widths. Note that the probabilities are spread out in the first iteration and it slowly converges to a starting point. They are still spread across the horizontal(width) axis because we vary the width only in a small range that is decided based on the amount of motion present in the sign.

extracting the sign. One could choose to run a multiple of the average number of times as
well, but we found the average number to be sufficient to show the stability of the solution in our experiments. Algorithm 5.3.1 presents the process as a pseudocode.

**Algorithm 5.3.1: EXTRACT SIGNEMES(L₁,⋯,Lₙ, A, B)**

**comment:** Generates multiple initialization vectors and calls ICM with each of them.

\[ N = \text{MEAN} (L_1, L_2, \cdots, L_n) \]

**for** \( j \leftarrow 1 \) **to** \( N \)

\[
\begin{aligned}
&\text{for } i \leftarrow 1 \text{ to } n \\
&\quad \text{do } \\
&\quad \text{do } w^0_i = \text{UNIFORM}(A \cdots B) \\
&\quad \quad a^0_i = \text{UNIFORM}(1 \cdots L_i - w^0_i + 1) \\
&\quad \quad \{a^j_i, w^j_i, \cdots, a^j_n, w^j_n\} = \text{ITERATED CONDITIONAL MODES}(a^0_1, w^0_1, \\
&\quad \quad \cdots, a^0_n, w^0_n) \end{aligned}
\]

**for** \( i \leftarrow 1 \) **to** \( n \)

\[
\begin{aligned}
&\text{comment: Assign most frequently occurring value as the final value.} \\
&\quad \text{do } w_i = \text{MODE}(w^j_i) \\
&\quad \quad a_i = \text{MODE}(a^j_i) \end{aligned}
\]

For extracting the sign ‘DEPART’ from 14 sentences, we had 89 frames per sentence on an average. Hence we ran 89 different ICM runs for extracting the common subsequence representing ‘DEPART’. Figure 5.5 shows the plots of histograms of start and end location of the sign in each of the 14 sentences from the 89 runs. It should be noted that in most of the sentences, more than 50% of the total number of runs result in the same solution.
Figure 5.5. Histograms showing the start and end locations of signs extracted from 14 different sentences using multiple ICM runs. The initial parameter vector for each ICM run was chosen independently using uniform random sampling. As it can be seen the start and end points found by most of the runs converge to the same solution (denoted by single high bars in most of sentences). The legend shown in the plot for the first sentence, $S_1$, holds for other sentences as well.

5.4 Results and Analyses

We test our approach of extracting signemes from both audio and video sequences representing sentences from American Sign Language. We describe the datasets here and presents the results obtained.
5.4.1 Datasets

For each dataset, we explain the features used for the common pattern extraction and give the number of sentences present in each of them.

5.4.1.1 Video Sequences

The video dataset consists of 136 American Sign Language (ASL) video sequences used to extract 10 common subsequences, using an average of 14 sequences to extract each sign. There are approximately 10 frames per sign. The data does not involve any color gloves or magnetic trackers. We perform skin-color segmentation to extract the skin blobs and compute relational distribution for each image. The relational distributions are embedded in the low dimensional SoRD space and each sequence is represented as a string of points in the SoRD space.

5.4.1.2 Audio Sequences

The audio dataset consists of spoken out sentences corresponding to the sequences in the video dataset described above. There were 136 continuous audio recordings used for extracting 10 words. Each recording consists of an ASL sentence spoken out in English, preserving the ASL syntax, such that the order of words in an audio recording corresponds to the sequence of signs signed in the continuous ASL sentences of the video dataset. The audio sequences are continuous in nature involving the natural co-articulation between the words. But the co-articulation in speech is far less than that in the sign language videos. The sequences are captured at 22 kHz. They are processed to extract 13 MFCCs at 25 frames per second. The extracted frames are projected onto a PCA space retaining all the 13 dimensions. PCA helps us get a better visualization of the sequences since the first
coefficient captures the highest variation among the data points. Each audio sequence is thus finally represented as a string of points in the PCA space.

5.4.2 Common Pattern Extraction

We perform extraction of the common patterns from a set of sentences from both audio and video datasets. As an example, Figure 5.6 depicts the result of extraction of the sign ‘DEPART’ from 14 video sequences. It plots the SoRD first dimension coefficients of the frames vs. the frame number for each sentence. The highlighted portions represent the signeme. The odd columns show the ground truth and the even columns show the corresponding results. As can be seen, the extracted patterns and the corresponding ground truth patterns very much similar, except for few frames at the beginning and end of some of the patterns. Note that since we deal with continuous video sequences, a difference of one or two frames between the ground truth and the extracted pattern is not considered as a problem.

Figure 5.7(a) shows the scatter plot of the ground truth start position vs. the estimated start position of the pattern extracted from each of the 136 sentences in the video dataset. Figure 5.7(b) shows the corresponding scatter plot for the end position of the patterns in the sentences. As can be seen most of the points in the scatter plots lie along the diagonal. It indicates that very few of the extracted patterns are wrong. Those correspond to the points lying far-off from the diagonal. All the extracted patterns were examined by a Sign Language expert. She had to recognize and label each of them as good, partially good or bad. Good indicates that both the start and end of the extracted pattern are correct. Partially good indicates that only one of them, i.e. either the start or the end is correct, and bad indicates that both the start and end points are incorrect. The results showed that out of the 136 extracted video patterns, 98 were considered good, 20 were partially good and 18 were bad. Figures 5.4.2 and 5.10 show instances of the good patterns extracted for
each sign. Figure 5.11 shows two *partially good* patterns and one *bad* pattern. We have uploaded all the extracted instances of different signs from our experiments to a webpage (See Appendix A). The sign ‘Passport’ had the largest number of *partially good* results. It is because the actual width of the sign in various instances varies more than the given width range. The *bad* extractions (e.g. Figure 5.11(c)) are often due to the presence of a second sign in the sentence that looked alike in the frontal view to the sign being extracted. Such problems can be addressed by using more discriminative features.

Figure 5.8(a) and (b) show the scatter plots of the ground truth vs. estimated position for the start and end positions respectively of the patterns extracted from 136 audio sequences in the audio dataset. As can be seen, almost all of the points except one, lie very close to the diagonal. This indicates that the start and end positions of the extracted subsequences coincide well with that of the ground truth common subsequences. We decide further accuracy by examining the individual audio clips and label them as *good*, *partially good* and *bad*, same as in the video dataset. Out of 136 extracted audio patterns, we had 127 *good* patterns, 8 *partially good* patterns and 1 bad pattern. All the extracted audio clips and the whole audio sentences can be found in the webpage described in Appendix A.
Figure 5.6. First dimension of the video sequences containing a common sign “DEPART”. The sequences are indicated by the dotted curves and the solid lines on each of them indicate the common pattern or signeme. The odd columns show the ground truth and the even columns show the results.
Figure 5.7. Extraction of common patterns or signemes from 136 video sequences. (a) and (b) show the scatter plots for the computed location vs. the ground truth location for the start and end points respectively. The closer the points are to the diagonal, the closer the result is to the ground truth.

Figure 5.8. Extraction of common patterns from 136 audio sequences. (a) and (b) show the scatter plots for the computed location vs. the ground truth location for the start and end points respectively. The closer the points are to the diagonal, the closer the result is to the ground truth.
Figure 5.9. Signemes extracted from sentences
Figure 5.10. Signemes extracted from sentences

- (a) PASSPORT - Partially Correct (The last frame shown does not belong to the sign)

- (b) CANT - Partially Correct (The last two frames do not belong to the sign)

- (c) CANT - Bad extraction (No part of the above extracted pattern belongs to the sign CANT)

Figure 5.11. Some partially correct and bad extractions

71
CHAPTER 6
CONCLUSIONS AND FUTURE WORK

In this dissertation, we developed and evaluated methods for efficiently representing articulated motion including signs, gestures and human activity sequences, without using any kind of tracking or correspondence between frames. Motion sequences of sign language, gesture or human activities are represented as a series of probability functions or relational distributions using relational attributes between low-level primitives in the images. These probabilities are embedded into a low dimensional configuration space retaining various probabilistic distance measures like Bhattacharya, Matusita, Chernoff, KL and Symmetric KL. Probability densities for new frames are efficiently embedded in the configuration space retaining the probabilistic distances from a set of reference points. The number of reference points needed to embed a new probability function depends both on the dimensionality of the space and the nature of the probabilistic distance measure. We also show that sign language motion representation using relational distributions is robust to the variation of edge parameters.

Current sign language researchers still use data gloves or magnetic trackers for across signers datasets. Relational distributions could be used for capturing motion across signers, which would rely on skin color segmentation of the hands and face, without the need of intrusive gloves. The changes in the relative configuration of the hands and face in the sequences can still be used for recognizing signs across signers. Richer features can also be added, if required, to the distributions to increase their discriminative powers.
Relational distributions could be used in other domains as well. For example, for automatic detection of motion or scene changes in the videos or for detecting key frames in the videos. They can also be used for organizing videos in video databases like You Tube, by classifying various videos based on the amount of motion involved in them. We show that random sampling provides a faster approach to compute relational distributions than the conventional exhaustive sampling. The random sampling approach along with our embedding framework makes it feasible to use relational distributions across multiple domains for representing, detecting or classifying motion patterns.

In this dissertation, we also formulated a probabilistic framework to automatically learn signs from continuous sign language sentences. Signs can be automatically extracted from color videos of multiple continuous sign language sentences concurrently. The extracted portions of the signs, called signemes, are robust to the variations of adjacent signs and the associated movement epenthesis. We also show results of our framework on an audio dataset by extracting common words from continuous spoken sentences. The extracted audio words could be used for audio annotation of signs in sign language sentences. Our signemes are at frame level, and hence they can also be used by other ASL researchers using different low-level features. Signs automatically extracted from continuous sentences can also be used for generating training datasets.

There are some ways we can further advance the ideas in this dissertation. The relational distributions used are only translation invariant and not rotation or scale invariant. While rotation invariance is often not desirable in gesture recognition, scale invariance is an important issue. One way to do that can be by normalizing the relational data by the standard deviation of the attributes before computing the histograms. Also, we used fixed size histograms for all the three datasets. Optimizing the histogram size depending on the dataset might further improve the results. Another issue is handling outliers. In the cases of extremely bad segmentation, e.g. a relevant body part is missed completely in some
frames or a large spurious blob appears in some frames, then the corresponding relational
distributions do not indicate the correct flow of motion. But it should still be possible to
recognize a test sequence if the frames of the training sequences are correctly segmented
and segmentation of a large majority of the frames in the test sequence is correct. The
outlier frames need to be detected and removed and the motion needs to be interpolated.

We addressed the extraction of only a single pattern from each sentence. It can be
extended to extract signemes that occur multiple times in the same sentence. Handling large
variations encountered when learning signemes across signers is also very challenging.
Conventional dynamic time warping cannot deal with amplitude variations in the signs
which is very common across signers. It can be extended to deal with amplitude differences
between various instances of the same sign.
REFERENCES


APPENDICES
Appendix A: Webpage with Common Pattern Extraction Results

We have uploaded all the common pattern extraction results to a webpage at http://figment.csee.usf.edu/~snayak/motifResults/SignemeExtraction.html. Figure A.1 shows it as screen shot. The webpage contains both the audio and video clips extracted. It also contains links to complete sentences from which the signemes were extracted. It should also be noted from the extracted signemes shown in the webpage that, in addition to locating the signeme correctly in sentences, our approach is also robust to the variations within the sign due to different contexts.

![Figure A.1](http://figment.csee.usf.edu/~snayak/motifResults/SignemeExtraction.html)
ABOUT THE AUTHOR

Sunita Nayak received Bachelor of Engineering degree in Computer Science and Engineering from National Institute of Technology, Rourkela, India in 2001 and Master of Science in Computer Science from University of South Florida in 2005. She has multiple publications and has made oral presentations in various conferences during her Ph.D. program at USF. Her research interests include image processing, computer vision and pattern recognition.