Automated Extraction of Signs from Continuous Sign Language Sentences using Iterated Conditional Modes

Sunita Nayak
Photometria Inc.
4320 La Jolla Village Dr.
San Diego, CA 92122
snayak@photometria.com

Sudeep Sarkar
Dept. of Computer Sc. & Engg.
University of South Florida
Tampa, FL 33620, USA
sarkar@cse.usf.edu

Barbara Loeding
Dept. of Special Education
University of South Florida
Lakeland, FL 33803, USA
bloeding@poly.usf.edu

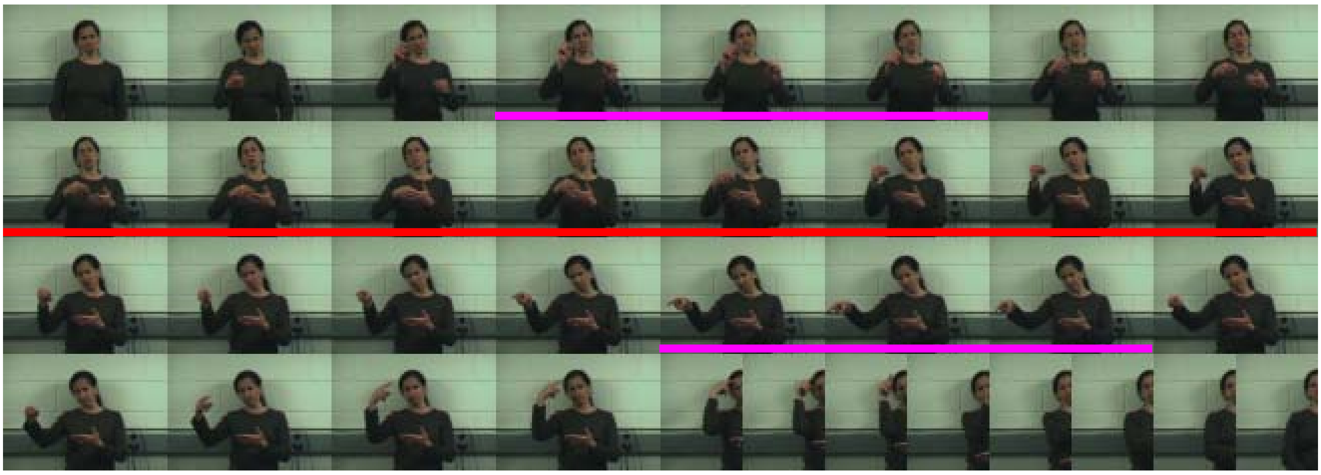
Abstract

Recognition of signs in sentences requires a training set constructed out of signs found in continuous sentences. Currently, this is done manually, which is a tedious process. In this work, we consider a framework where the modeler just provides multiple video sequences of sign language sentences, constructed to contain the vocabulary of interest. We learn the models of the recurring signs, automatically. Specifically, we automatically extract the parts of the signs that are present in most occurrences of the sign in context. These parts of the signs that is stable with respect to adjacent signs, are referred to as signemes. Each video is first transformed into a multidimensional time series representation, capturing the motion and shape aspects of the sign. We then extract signemes from multiple sentences, concurrently, using Iterated Conditional Modes (ICM). We show results by learning multiple instances of 10 *different signs from a set of* 136 *sign language sentences. We classify the extracted signemes as* correct, partially correct *or* incorrect *depending on whether both the start and end locations are correct, only one of them is correct or both are incorrect, respectively. Out of the* 136 *extracted video signemes,* 98 *were correct,* 20 *were partially correct and* 18 *were incorrect. To demonstrate the generality of the unsupervised modeling idea, we also show the ability to automatically extract common spoken words in audio. We consider the English glosses (spoken) corresponding to the sign language sentences and extract the audio counterparts of the signs. Of the* 136 *such instances, we recovered* 127 *correct,* 8 *partially correct, and* 1 *incorrect representation of the words.*

1. Introduction

Sign language research in the vision community has primarily focused on improving recognition rates of signs either by improving the motion representation and similarity measures [25, 2, 23, 4, 14] or by adding linguistic clues during the recognition process [6, 9]. Ong and Ranganath [18] presented a review of the automated sign language research and also point out one important issue in continuous sign language recognition. While signing a sentence, there exist transitions of the hands between two consecutive signs that do not belong to either of the signs. This is called movement epenthesis [15]. This needs to be dealt with first before dealing with any other phonological issues in sign language [18]. Most of the existing works in sign language assume that the training signs are already available and often signs used in the training set are the isolated signs with the boundaries chopped off, or manually selected frames from continuous sentences. Unlike isolated signs, a sign in a continuous sentence is strongly affected by its context in the sentence. Figure 1 shows a continuous sentence 'YOU CAN BUY THIS FOR HER'. The frames representing the sign 'BUY' and the neighboring signs are marked. The unmarked frames between the signs indicate the frames corresponding to movement epenthesis, which depends on the end and start of the preceding and succeeding sign respectively. The movement epenthesis also affects how the sign is signed. This effect makes the automated learning and recognition of signs from continuous sentences harder than isolated signs, fingerspelling or plain gestures.

In this paper, we address the problem of automatically extracting the part of a sign that is most common in all occurrences of the sign, and hence expected to be robust with respect to the variation of adjacent signs. These common parts can be used for spotting or recognition of signs in continuous sign language sentences using either Hidden Markov Models or Dynamic Time warping. They can also be used by sign language experts for teaching or studying variations between instances of signs in continuous sign language sentences, or in automated sign language tutoring systems. Further, they can be used even in the process of



(a) Continuous Sentence 'YOU CAN BUY THIS FOR HER'

Figure 1. Movement epenthesis in sign language sentences. The frames corresponding to the sign 'BUY' are marked in red. The adjacent signs in each sentence are marked in magenta. The frames in between the marked frames represent movement epenthesis i.e. the transition between signs.

translating sign language videos directly to spoken words.

A different but a closely related problem is the extraction of common subsequences, also called motifs, from very long multiple gene sequences in biology [3, 13]. But due to the univariate and discrete nature of the biological sequences, their algorithms are not always directly applicable to other multivariate continuous domains in time series like speech or sign language. Some of the motif discovery works illustrating results on human movements include [10, 16, 1]. Nayak *et al.* [17] find recurrent patterns from sign language sentences. Yang *et al.* [24] perform sign spotting in continuous sign language sentences using Conditional Random Fields. Farhadi *et al.* [11] used models learned using an avatar to spot signs signed by a human signer.

Following the success of Hidden Markov Models (HMMs) in speech recognition, they were used by sign language researchers [22, 20, 6, 4, 21] for representing and recognizing signs. But HMMs need a large number of training data and unlike speech, data from native signers is not yet as easily available as speech data. Our current work can be used to automatically generate such training data, cutting down on the tedious nature of the process. Nayak et al. [17] proposed a sequential method for extracting signeme models from continuous sign language sentences. But the signemes extracted were heavily biased by the first two sentences used to start the search. They matched each substring of a fixed length from an interpolated sequence of the first sentence to every substring of the same length in the interpolated sequence in the second sequence. The best matching pair of substrings was considered to represent the common pattern and one of those patterns was used to extract the similar patterns from the rest of the sentences. Our work in this paper is different in multiple respects. Firstly, we present an approach to *simultaneously* extract the signemes from all given sentences. Secondly, our framework also accommodates the comparison of substrings of different widths. The set of sequences can be either in interpolated (speed-normalized) or in non-interpolated forms.

In this paper, we present a Bayesian framework to extract the common subsequence i.e. signeme from all the given sentences simultaneously. We assume that the sign to be extracted is the only sign that is common to each of the n given sentences. Skin color blobs are extracted from frames of color video, and a *relational distribution* [17] is formed for each frame using the edge pixels in the skin blobs. Each sentence is then represented as a trajectory in a low dimensional space called Space of Relational Distributions, which is arrived at by performing Principal Component Analysis (PCA) on the relational distributions. The trajectory implicitly captures the shape and motion in the video. The starting locations and widths of the candidate signemes in all the n sentences are together represented by a parameter vector. The initial values for the starting locations are obtained using uniform random sampling on the sequence of each sentence, and the 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 [19, 12] like the Gibbs sampling [7], which is a special case of the Metropolis-Hastings algorithm [8] 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 paper, we adopt a greedy approach based on the use of Iterated Conditional Modes (ICM) [5]. ICM converges much faster than a Gibbs sampler, but is known to be largely dependent on the initialization. We mitigate against this limitation by performing ICM a number of times equal to the average length of the n sentences, with different initializations. This strategy gave us good results with real data. The most frequently occurring solution from all the ICM runs is considered as the final solution.

We test our algorithm by extracting 10 different signs from 136 sentences, using 14 sentences on an average to extract each sign. To demonstrate the generality of the modeling approach, we also show results on audio data. The audio data consists of spoken English glosses corresponding to the video data. Each audio sequence is represented as a multi-dimensional time series in a PCA space. The extracted common words (spoken) can be used to construct an audio-video dictionary of the signemes. An alternative way to create this audio-video dictionary, would be to use commercial speech recognition software on the English glosses and then use simple common text finding routines to label the signs. However, this is not an option, since commercial speech recognition systems are trained on English grammar and ASL grammar is not the same as that of spoken English.

The contributions of this paper can be summarized as follows: (i) we present an unsupervised approach to automatically extract parts of signs that are robust to the variation of adjacent signs, simultaneously from multiple sign language sentences, (ii) our approach does not consider all possible parameter combinations, instead samples each of them in a sequential manner till convergence, which saves a lot of computation, and (iii) we show results on extracting signs from plain color videos of continuous sign language sentences without using any color gloves or magnetic trackers, and also on audio data.

We organize the paper as follows. Section 2 formulates the problem of finding signemes from a given set of long sequences in a probabilistic framework. We describe how we solve it using Iterated Conditional Modes. It is then followed by a description of our experiments and results in Section 3. Finally, Section 4 concludes the paper and discusses possible future work.

2. Problem Formulation

Sign language sentences are series of signs with movement epenthesis in between signs. Signeme represents the portion of the sign that is most similar across the sentences [17]. 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 similar 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, one from each of the n sentences, and θ_m denote the parameter set representing the target set of signemes in the n sentences. We find θ_m using a probabilistic framework.

$$\theta_m = \arg\max_{\theta} p(\theta) \tag{1}$$

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:

$$p(\theta) = \frac{g(\theta)}{\sum_{\theta} g(\theta)} \tag{2}$$

where

$$g(\theta) = \exp\left(-\beta \sum_{i=1}^{n} \sum_{j=1}^{n} d(\vec{s}_{a_i}^{w_i}, \vec{s}_{a_j}^{w_j})\right) \tag{3}$$

and β is a positive constant. Note that $g(\theta)$ varies inversely with the summation of the pair-wise distances of all the subsequences given by θ. Also note that $p(\theta)$ is hard to compute or even sample from because it is computationally expensive to compute the denominator in Eq. 2, as it involves the summation over all possible parameter combinations. β 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 [7] are used in later steps, then the rate of convergence would be determined by this parameter.

Let θ_i represent the parameters from the i^{th} sentence, i.e. $\{a_i, w_i\}$ and $\theta_{(i)}$ represent the rest of the parameters, $\{a_1, w_1 \cdots a_{i-1}, w_{i-1}, a_{i+1}, w_{i+1} \cdots a_n, w_n\}$. To make sampling easier, we construct *conditional* density function of the parameters from each sentence, i.e. θ_i, *given* the values of the rest of the parameters, i.e. $\theta_{(i)}$. 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. 2.

$$f(\theta_i|\theta_{(i)}) = \frac{p(\theta)}{p(\theta_{(i)})} = \frac{p(\theta)}{\sum_{\theta_i} p(\theta)} = \frac{g(\theta)}{\sum_{\theta_i} g(\theta)} \tag{4}$$

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

$$f(\theta_i|\theta_{(i)}) = \frac{\exp\left(-\beta \sum_{k=1}^{n} d(\vec{s}_{a_i}^{w_i}, \vec{s}_{a_k}^{w_k})\right)}{\sum_{\theta_i} \exp\left(-\beta \sum_{k=1}^{n} d(\vec{s}_{a_i}^{w_i}, \vec{s}_{a_k}^{w_k}))\right)} \quad (5)$$

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:

$$f(\theta_i|\theta_{(i)}) = \frac{g(\theta_i|\theta_{(i)})}{\sum_{\theta_i} g(\theta_i|\theta_{(i)})} \quad (6)$$

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

2.1. 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:

$$\begin{aligned} D(i, j) =& e(i, j) + \\ & \min \{D(i-1, j), D(i-1, j-1), D(i, j-1)\} \end{aligned} \quad (7)$$

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.

2.2. Parameter Estimation

In order to extract the common sign from a given set of sign language sentences, we need to compute θ_i for each of the sentences sequentially. Gibbs sampling [7] is a Markov Chain Monte Carlo approach [12] 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 [5]. 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 2.1: ITERATED CONDITIONAL MODES($\{a_1^0, w_1^0, \cdots, a_n^0, w_n^0\}$)

```
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:
$\theta_0 \leftarrow \{a_1^0, w_1^0, \cdots, a_n^0, w_n^0\}$
repeat
  for $i \leftarrow 0$ to $n$
    do
      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 $g(a_i, w_i|\theta_{(a_i,w_i)}) \leftarrow \exp\left(-\beta \sum_{k=1}^{n} d(\vec{s}_{a_i}^{w_i}, \vec{s}_{a_k}^{w_k})\right)$
      comment: Normalize
      for $w_i \leftarrow A$ to $B$
        do
          for $a_i \leftarrow 0$ to $L_i - w_i + 1$
            do $f(a_i, w_i|\theta_{(a_i,w_i)}) \leftarrow \frac{g(a_i,w_i|\theta_{(a_i,w_i)})}{\sum_{a_i,w_i} g(a_i,w_i|\theta_{(a_i,w_i)})}$
      $a_i, w_i \leftarrow$ 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 2.1 outlines the process of ICM to extract 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

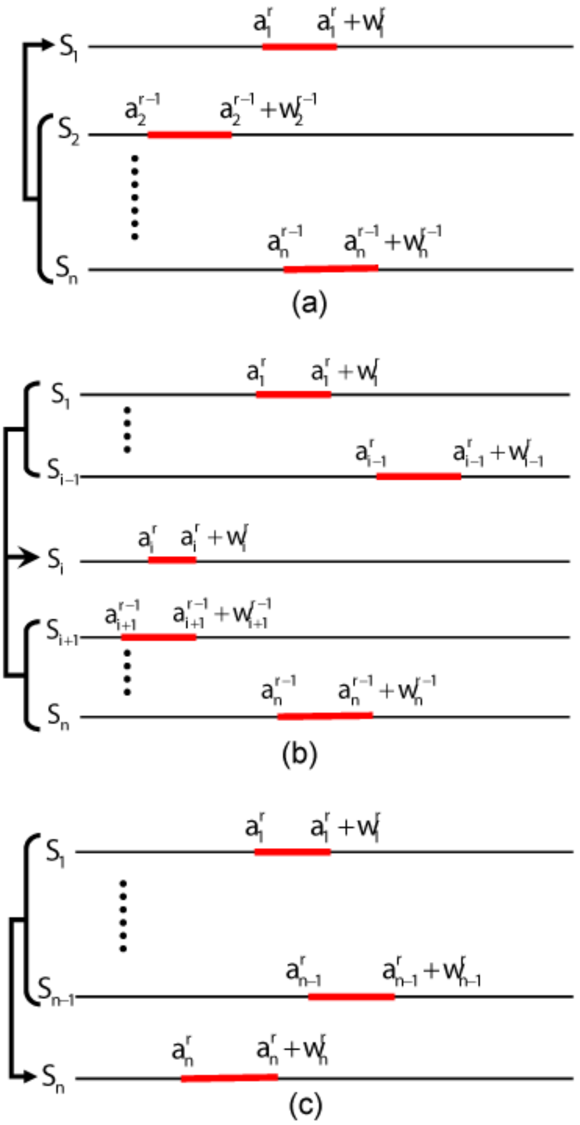


Figure 2. Sequential update of the parameter values using ICM. (a), (b) and (c) respectively show the parameter updates in the first sentence, the i^{th} and the n^{th} sentences. In the r^{th} iteration, the parameters of the common sign in i^{th} sentence is computed based on the parameter values of the previous $(i-1)$ sentences obtained in the same iteration, and those of the $(i+1)^{th}$ to n^{th} sentences obtained in the previous, i.e. the $(r-1)^{th}$ iteration.

any more. Figure 2 depicts the sampling process for a single iteration, r. Note the conditional and sequential nature of sampling from various sentences within the single iteration.

2.3. 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 θ. The width w_i^0 is obtained by sampling a width value based on uniform random distribution from the set of all possible widths in a given range $[A, B]$. The value for a_i^0 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 $\{1 \cdots (L_i - w_i^0 + 1)\}$ where L_i is the length of the sequence S_i. 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 extracting the sign. For example, for extracting the sign 'DEPART' from 14 sentences with an average of 89 frames per sentence, we ran 89 different ICM runs. 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 2.2 presents the process as a pseudocode.

Algorithm 2.2: EXTRACT SIGNEMES$(L_1, \cdots L_n, A, B)$

```
comment: Generates multiple initialization vectors and
calls ICM with each of them.
$N \leftarrow$ MEAN$(L_1, L_2, \cdots, L_n)$
for $j \leftarrow 1$ to $N$
  do
    for $i \leftarrow 1$ to $n$
      do
        $w_i^0 \leftarrow$ UNIFORM$(A \cdots B)$
        $a_i^0 \leftarrow$ UNIFORM$(1 \cdots L_i - w_i^0 + 1)$
    $\{a_1^j, w_1^j, \cdots, a_n^j, w_n^j\} \leftarrow$
          ITERATED CONDITIONAL
          MODES$(a_1^0, w_1^0, \cdots, a_n^0, w_n^0)$
for $i \leftarrow 1$ to $n$
  do
    comment: Assign most frequently occurring value
    as the final value for each parameter.
    $w_i \leftarrow$ MODE$(w_i^j)$,   $a_i \leftarrow$ MODE$(a_i^j)$
```

3. Experiments and Results

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

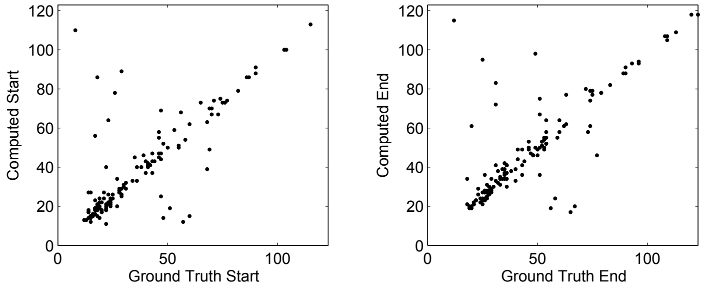
3.1. Datasets

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 a relational distribution for each image. The relational distributions are embedded in the low dimensional Space of Relational Distributions(SoRD) space using Principal Component Analysis. Each sequence is then represented as a string of points in the SoRD space.

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.

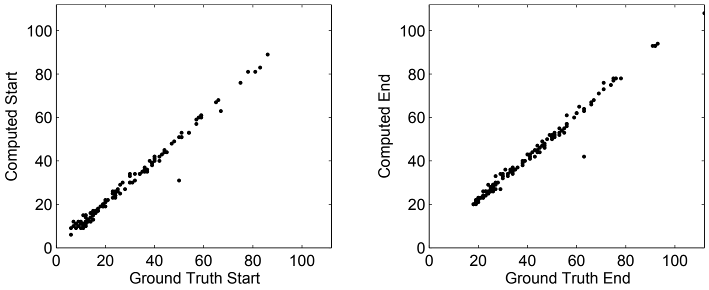
3.2. Common Pattern Extraction Results

Figure 3(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 3(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 incorrect. Those correspond to the points lying far-off from the diagonal. All the extracted patterns were examined by an American Sign Language expert. She had to recognize and label each of them as *correct, partially correct* or *incorrect*. *Correct* indicates that both the start and end of the extracted pattern are correct. *Partially correct* indicates that only one of them, i.e. either the start or the end is correct, and *incorrect* indicates that both the start and end points are incorrect. The results showed that out of the 136 extracted video patterns, 98 were considered *correct*, 20 were *partially correct* and 18 were *incorrect*. Figure 5 show one instance of the signeme extracted from each sign. We have linked all the instances of different signs extracted from our experiments to a web page at http://marathon.csee.usf.edu/ASL/SignemeExtraction.html. The web page also contains links to complete sentences from which the signemes were extracted. It should also be noted from the extracted signemes 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.



(a) Video Start Point Estimation (b) Video End Point Estimation

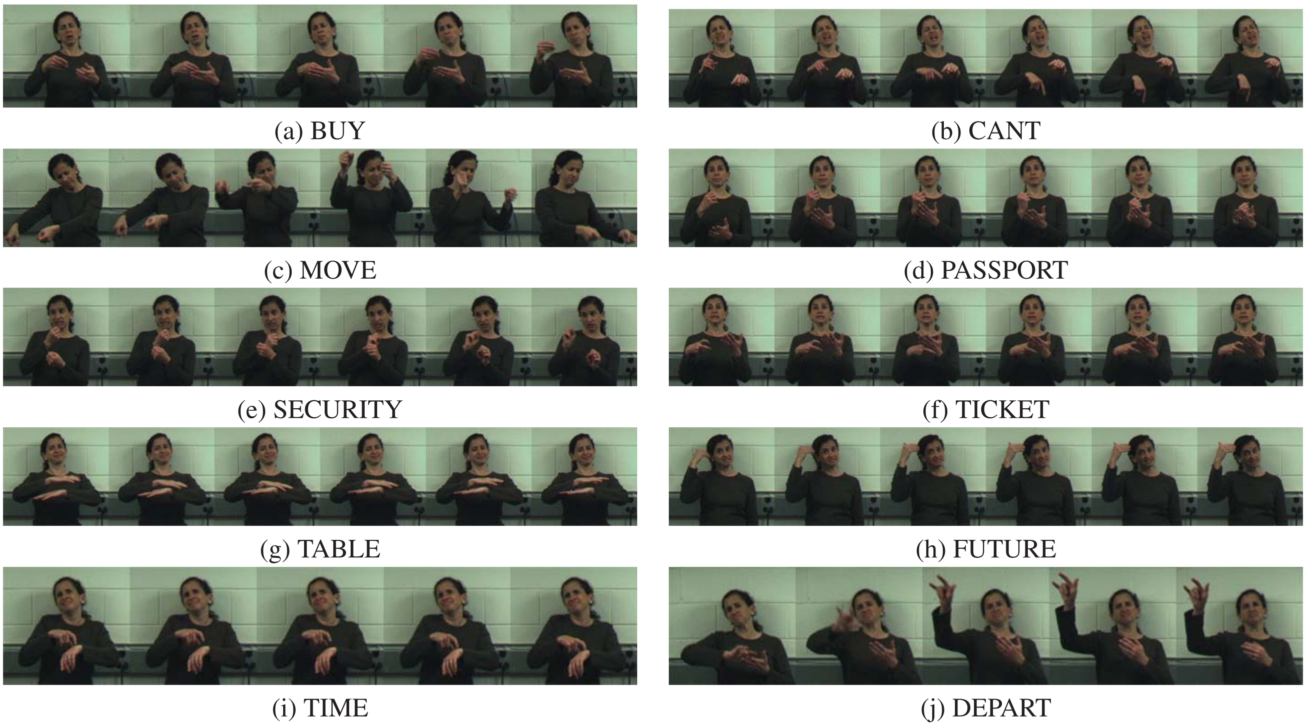
Figure 3. Extraction of common patterns or signemes from 136 video sequences. (a) and (b) show the scatter plots for the computed location vs. the ground truth location for the start and end points respectively. The closer the points are to the diagonal, the closer the result is to the ground truth.



(a) Audio Start Point Estimation (b) Audio End Point Estimation

Figure 4. Extraction of common patterns or signemes 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 4(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. 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 labeling them as *correct, partially correct* and *incorrect*, same as in the video dataset. Out of 136 extracted audio patterns, we had 127 *correct* patterns, 8 *partially correct* patterns and 1 *incorrect* pattern. All the extracted audio clips and the whole audio sentences can be found in the



(a) BUY

(b) CANT

(c) MOVE

(d) PASSPORT

(e) SECURITY

(f) TICKET

(g) TABLE

(h) FUTURE

(i) TIME

(j) DEPART

Figure 5. Signemes extracted from sentences. Note that for display purposes some of the intermediate frames have been dropped.

same web page as the video clips.

4. Conclusion and Future Work

We presented a novel algorithm to extract signemes, i.e. the common pattern representing a sign, from multiple long video sequences of American Sign Language. A signeme is a part of the sign that is robust to the variations of the adjacent signs and the associated movement epenthesis. We first represent each sequence as a series of points in a low dimensional Space of Relational Distributions, and then use a probabilistic framework to locate the signemes in each sequence concurrently. We use Iterative Conditional Modes (ICM) to sample the parameters, i.e. the starting location and width of the signeme in each sentence in a sequential manner. In order to overcome the local convergence problem of ICM, we run it repetitively with uniformly and independently sampled initialization vectors. The number of times ICM is repeated depends on the average length of the sequences used for extracting the signeme. We show results on ASL video sequences that do not involve any magnetic trackers or gloves, and also on a corresponding audio dataset. The extracted signemes also show that our approach is robust to some extent to the variations produced within a sign due to different contexts.

The approach in this paper can be used to speed up training set generation for ASL algorithms by drastically reducing the manual aspect of the process. Instead of the current need of manually demarcated signs in continuous sentences, we would just need instances of sentences containing the sign whose model is sought. The sentences do not have to be associated with any English glosses, either. The current scope of the algorithm is limited to single instance in each sentence of the sign that is being modeled. However, the ICM-based approach can be extended to multiple instances too, of course, with the added expansion of the search dimensionality. This is one possible direction for future work. Another contribution of this work is an empirically derived robust representation of the sign that is stable with respect to the variations due to neighboring signs and sentence context. While it is not clear if such representations has linguistic validity, these stable representations could be useful for detection or spotting of signs and gestures in extended gesture sequences.

5. Acknowledgment

This work was supported by funding from US National Science Foundation (NSF) ITR Grant IIS 0312993.

References

[1] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic. Discovering clusters in motion time-series data. *Computer Vision and Pattern Recognition*, pages 375–381, 2003.

[2] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: a method for efficient approximate similarity rankings. *Computer Vision and Pattern Recognition*, page 268275, 2004.

[3] T. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21:51–80, 1995.

[4] B. Bauer and H. Hienz. Relevant features for video-based continuous sign language recognition. In *Automatic Face and Gesture Recognition*, page 440445, 2000.

[5] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, pages 259–302, 1986.

[6] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *European Conference on Computer Vision*, volume 1, page 390401, 2004.

[7] G. Casella and E. George. Explaining the Gibbs sampler. *The American Statistician*, 46:167–174, 1992.

[8] S. Chib and E. Greenberg. Understanding the MetropolisHastings algorithm. *The American Statistician*, 49:327335, 1995.

[9] K. Derpanis, R. Wildes, and J. Tsotsos. Hand gesture recognition within a linguistics-based framework. *Euro. Conf. on Computer Vision*, pages 282–296, 2004.

[10] F. Duchne, C. Garbay, and V. Rialle. Learning recurrent behaviors from heterogeneous multivariate time-series. *Arificial Intelligence in Medicine*, (1):25–47, 2007.

[11] A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[12] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall, 1998.

[13] C. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[14] J. F. Lichtenauer, E. A. Hendriks, and M. Reinders. Sign language recognition by combining statistical dtw and independent classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:2040–2046, November 2008.

[15] S. Liddell and R. Johnson. American Sign Language: The phonological base. *Sign Language Studies*, pages 195–277, 1989.

[16] D. Minnen, C. Isbell, I. Essa, and T. Starner. Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. *Conference on Artificial Intelligence*, 2007.

[17] S. Nayak, S. Sarkar, and B. Loeding. Unsupervised modeling of signs embedded in continuous sentences. *IEEE Workshop on Vision for Human-Computer Interaction*, 2005.

[18] S. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:873–891, June 2005.

[19] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 2004.

[20] T. Starner and A. Pentland. Real-time American Sign Language recognition from video using Hidden Markov Models. *Computational Imaging and Vision*, 9:227244, 1997.

[21] T. Starner, J. Weaver, and A. Pentland. Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[22] C. Vogler and D. Metaxas. Parallel Hidden Markov Models for American Sign Language Recognition. *International Conference on Computer Vision*, 1:116, 1999.

[23] Q. Wang, X. Chen, L. Zhang, C. Wang, and W. Gao. Viewpoint invariant sign language recognition. In *Computer Vision and Image Understanding*, volume 108, pages 87–97, 2007.

[24] H. Yang, S. Sclaroff, and S. Lee. Sign language spotting with a threshold model based on conditional random fields. *Accepted in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[25] M. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1061–1074, 2002.