Enhanced Level Building Algorithm for the Movement Epenthesis Problem in Sign Language Recognition

Ruiduo Yang[1], Sudeep Sarkar[1], and Barbara Loeding[2]
[1]Computer Science and Engineering
University of South Florida
Tampa, FL 33620, USA
{ryang, sarkar}@csee.usf.edu
[2]Special Education
University of South Florida
Lakeland, FL 33603
bloeding@lklnd.usf.edu

Abstract

*One of the hard problems in automated sign language recognition is the movement epenthesis (*me*) problem. Movement epenthesis is the gesture movement that bridges two consecutive signs. This effect can be over a long duration and involve variations in hand shape, position, and movement, making it hard to explicitly model these intervening segments. This creates a problem when trying to match individual signs to full sign sentences since for many chunks of the sentence, corresponding to these* me*s, we do not have models. We present an approach based on version of a dynamic programming framework, called Level Building, to simulataneously segment and match signs to continuous sign language sentences in the presence of movement epenthesis (*me*). We enhance the classical Level Building framework so that it can accomodate* me *labels for which we do not have explicit models. This enhanced Level Building algroithm is then coupled with a trigram grammar model to optimally segment and label sign language sentences. We demonstrate the efficiency of the algorithm using a single view video dataset of continuous sign language sentences. We obtain 83% word level recognition rate with the enhanced Level Building approach, as opposed to a 20% recognition rate using a classical Level Building framework on the same dataset. The proposed approach is novel since it does not need explicit models for movement epenthesis.*

1. Introduction

The task of sign language recognition offers an unique opportunity for the development of motion recognition algorithms for human computer interfaces. In particular, it lets us easily get beyond just single gestures or signs. In practice HCI would involve composition of individual gestures just as sign sentences are compositions of individual signs. When signs appear in sentence contexts, variations

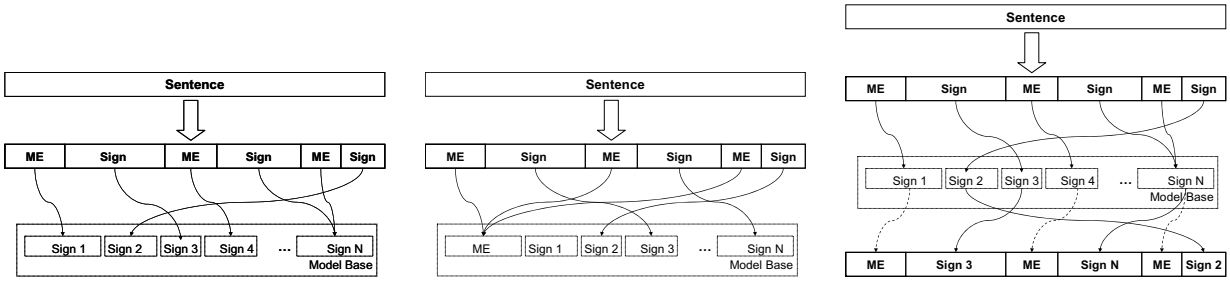
Figure 1. The first frame is the end of sign:"GATE", the last frame is the start frame of "WHERE", in between there are several transition frames which actually has no meaning and is known to be the me segment.

appear; sentences are not the concatenation of individual signs.

In the phonological processes in sign language, sometimes a movement segment needs to be added between two consecutive signs [10]. This is called movement epenthesis (me). Fig. 1 shows an example of me frames. These frames do not correspond to any sign and can involve change in hand shape, movement, and can be over many frames sometimes equal in length to actual signs. Given N possible signs there would be $\mathcal{O}(N^2)$ possible types of movement epenthesis, which make it computationally burdensome to explicitly model all possible mes. There are also other types of phonological processes where the appearance of a sign is affected by the previous and successive signs; these processes include hold deletion, metathesis and assimilation. These are analogous to the "coarticulation" issue in speech [4]. There is no correlate for "movement epenthesis" in speech. Movement epenthesis occurs very frequently between consecutive signs, unlike the "coarticulation"-like processes, which only occur in a small number of signs [2]. We concur with Sylvie and Ranganath [2], that the movement epenthesis should be dealt with first. The new matching algorithm in this paper is a contribution in that direction.

As one can easily garner from an excellent review of sign language recognition [2] Hidden Markov Models (HMM) [8] with statistical grammar modeling or Dynamic Time Warping (DTW) [7] approach are the most common ones. Both originate from the speech recognition community, where it has been found that the performance of both approaches are simlar. Since movement epenthesis is not

1-4244-1180-7/07/$25.00 ©2007 IEEE



(a) Ignoring me (b) Explicitly Modeling me (c) Our approach

Figure 2. Different approaches to handling movement epenthesis (me) in sentences: (a) If the effect of me is ignored while modeling, this will result in some me frames to be falsely classified as signs (b) If me is explicitely modeled, building such models will be difficult when the vocabulary grows large, since phonemes models are not well established for sign languages. (c) The adopted approach in this paper do not explicitly model mes instead, we allow for the possibility for me to exist when no good matching can be found.

a problem in speech, these approaches do not handle. One possibility is to use a grammar framework to resolve errors, however, only some of the errors can be corrected in that manner. We will show some results later.

Probably the first work in sign language recognition that addressed the me problem is by Vogler and Metaxas [11]. They explicitly modeled movement epenthesis in a continuous sign language recognition system with dedicated HMMs. They also experimented with modeling movement epenthesis and signs together as context-dependent signs [12]. Likewise, Yuan et.al [14] and Gao et.al [3] also explicitly modeled movement epenthesis and performed matching with both the sign and movement epenthesis models. The difference was that they adopted an automatic approach to cluster movement epenthesis frames in the training data first. Other than this, Yang and Sarkar [13] use conditional random fields to segment sentence by removing me segments, but their approach do not produce a recognition result. While explicit modeling of movement epenthesis have been shown to yield improved results, concerns of scalability of that approach remain; the number of possible me's is quadratically related to the number of signs in the dataset. Demands on the training dataset size would also increase significantly. Currently, we are not aware of any dataset that will enable us to do this in a statistcially meaningful manner. One way to overcome this demand for large training dataset could be to use an analgous concept to "phonemes"in speech, thus reducing the set of possible units to model. However, the concept of phonemes is not yet an well established concept for sign languages.

In this work, we present a novel sign recognition strategy that do not require explicit modeling of movement epenthesis. The basic difference of our approach with others is illustrated in Fig. 2. Fig. 2(a) represents a matching procedure that completely ignores mes. In such cases movement epenthesis frames between two signs can be recognized as a sign, leading to insertation errors hard to resolve even using grammar models. Fig. 2(b), on the other hand, shows the process of explicitly modeling movement epenthesis(me), where the me frames in the test sequence can be matched to the modeled me frames, not a sign. Fig. 2(c) illustrates our approach that does not use explicit me models. We just have a model base of signs to be recognized, but not movement epentheses.

We enhance the classical Level Building (LB) algorithm [7], based on dynamic programming approach, to match without explicit me models. The classical LB algorithm itself cannot handle the issue of me without explicit models. While the searching of the optimal signs sequence, we dynamically decide whether a match is a good match. If not, we allow for the me-label. This "me" label usually happens between two actual signs, and it contributes no meaning to the final result. Errors are further reduced by coupling the level building process with a trigram grammar model as a constraint. One of the byproducts of our approach is the segmentation of the sentence into signs and me chunks. The advantage of the approach is the reduced demands on video-based training data. Note that the trigram grammar model can be constructed from a sign language text corpus, without associated video. Although we demonstrate a deterministic, dynamic programming approach, the framework can be easily extended to a probabilistic framework, such as HMMs, as is done classically.

We conducted our experiments on a video sign language database with single front view, necessiating the need for extraction and representation of low-level features. We will outline a hand segmentation approach based on key frames and adopt a histogram based representation, described later, as features. The hand segmentation approach utilizes both motion and skin cues. It utilizes a set of keyframes to model the background. The detection algorithm exploits the fact that the hand is changing its representation (location and shape) faster than other parts of the scene (including face) during signing.

Since the core algorithmic contribution of this paper is the enhanced Level Building (eLB) approach, we will present it first, followed by the low-level segmentation and representations in Section 5. We start the presentation of the eLB algorithm by formally presenting the underlying problem in the next section.

2. Problem Formulation

Let the set of V model signs in the training database be represented as:

$$S_i = \langle s_i^1, s_i^2, \cdot, s_i^{N_i} \rangle \tag{1}$$

where $1 \leq i \leq V$, and N_i is the number of frames in the ith sign model.

In addition to these signs, we will use symbols to represent movement epenthesis `me` labels of various lengths. We, of course, do not have explicit models corresponding to these symbols. We use these symbols for the convinience of expressing the problem mathematically.

$$S_{V+k} = \langle c^1, c^2, \cdots, c^k \rangle \tag{2}$$

where $1 \leq k \leq N_{max}$ and N_{max} is the maximum `me` length. $c^1, \cdots, c^k$ are the dummy frames in the `me` labeled signs.

Let the test sequence T of length M be denoted by:

$$T = \langle t^1, t^2, \cdots, t^M \rangle \tag{3}$$

Our objective is to find S_e^* in the set of all the candidate signs sequences S^* such that the distance between S_e^* and T is minimized. We represented S^* as:

$$S^* = \{S_1^*, S_2^*, \cdots, S_{N^*}^*\} \tag{4}$$

where

$$S_i^* = \langle S_{q_i^1}, S_{q_i^2}, \cdots, S_{q_i^{L_i}} \rangle \tag{5}$$

where L_i denotes the number of signs in S_i^*, N^* denotes the number of all candidate signs sequences.

The subcripts $q_i^1, q_i^2, \cdots, q_i^{L_i}$ are such that $1 \leq q_i^1, q_i^2, \cdots, q_i^{L_i} \leq N$. We use L_{max} to denote the maximum number of signs one sentence can have.

We need to find S_e^* such that

$$e = \arg\min_i D(S_i^*, T) \tag{6}$$

where $D(.)$ is the function to compute distance score.

In terms of dynamic warping term, we seek an optimal path to match T and the candidate sign *sequences* in order to compute the distance scores. Mathematically, considering the optimal warping path $P(u)$ as a multi-valued function such that

$$P(u) = (T(u), S(u)) \tag{7}$$

9 where $1 \leq u \leq N_u$ is the index, N_u is the length of the warping path, $(T(u), S(u))$ represents the sequence of coordinates of the warping path, that is, the $T(u)$th frame of the test sequence is matched with the $S(u)$th frame of the candidate sign sequences S_i^*. $S(u)$ can be represented as the combination of a sign coordinate and a sub-sign coordinate such as:

$$S(u) = (Q(u), K(u)) \tag{8}$$

that is, the $T(u)$th frame of the test sequence is matched with the $K(u)$th frame in the $Q(u)$th sign in the candidate signs sequence. Hence Eq. 6 can be rewritten as:

$$e = \arg\min_i \sum_{u=1}^{N_u} d(t^{T(u)}, s_{i^*}^{K(u)}), i^* = q_i^{Q(u)} \tag{9}$$

The function $d(.)$ is the distance between a test frame and a frame from the model sequences, included the dummy `me` symbols. For distances with the frames in the V model signs this would depend on the choice of the low-level features and the distance measure used. We denote this by $M(t^i, s_j^k)$. The cost of a `me` label is denoted by α.

$$d(t^i, s_j^k) = \begin{cases} M(t^i, s_j^k), & \text{if } j \leq V \\ \alpha, & \text{if } j > V \end{cases} \tag{10}$$

The use of the `me` label cost, α, is the essential difference between the classical problem formulation for recognizing connected words in speech and our formulation for recognition of connected signs in sign languages. The choice of α, which is quite important, will be discussed later.

3. The Enhanced Level Building Algorithm

One naive way to obtain the solution of Eq. 9 is to enumerate among all the possible `sign sequence` candidates S_i^*, compute the warping distance score between S_i^* and T, find the S_i^* with minimum score. Clearly the computational complexity of such an approach is prohibitive. Hence, we adopt an sequential approach to build this optimal sign sequence using a framework called Level Building and enhance it to allow for `me` labels.

Each level corresponds to the possible order of signs or `me` in the test sentence. Thus, the first level is concerned with the first possible label in the sentence, and so on. Each level is associated with a set of possible start and end locations within the sequence. And at each level we store the best possible match for each combination of end point from the previous level. The optimal sequence of signs and `me` labels is constructed by backtracking.

For each level l, we store the optimal cost for matching between sign S_i and with the ending frame as m using a 3 dimensional array A:

$$A_l^i(m), \quad 1 \leq l \leq L_{max}, 1 \leq i \leq N, 1 \leq m \leq M \tag{11}$$

where

$$A_l^i(m) = \begin{cases} D(S_i, T_1^m), & \text{if } l = 1 \\ \min_{k,j} A_{l-1}^k(j) + D(S_i, T_{j+1}^m) & otherwise \end{cases} \quad (12)$$

T_j^m denotes a subsequence of the test sequence that starts at the jth frame and ends at the mth frame. Hence $A_l^i(m)$ gives us the minimum cumulative score for matching the ith model sign, S_i to the test sequence upto the m-th frame, for the lth sign label in the sequence. For distances to me labels we use

$$D(S_i, T_{j+1}^m) = (m - j)\alpha, \quad i > V \quad (13)$$

On the other hand, for an actual sign, the dynamic time warping score is used.

To enable us to reconstruct the sign sequence by backtracking, we use a predecessor array ψ, whose indices correspond to A.

$$\psi_l^i(m), \quad 1 \le l \le L_{max}, 1 \le i \le N, 1 \le m \le M \quad (14)$$

where

$$\psi_l^i(m) = \begin{cases} -1, & \text{if } l = 1 \\ \arg\min_k A_{l-1}^k(j) + D(S_i, T_{j+1}^m) & otherwise \end{cases} \quad (15)$$

The optimal matching score D^* is:

$$D^* = \min_{l,i} A_l^i(M) \quad (16)$$

To obtain the optimal signs sequence, we need to do backtracking according to ψ.

Fig 3 illustrates us the matching process of the enhanced LB algorithm. At end of each level we could obtain the best matched sequences. Note all the signs S_{v+k} is actually a me segment with k frames, where we do not have any models associating with them.

3.1. Grammar Constraint

The explorations at each level can be constrained by statistical grammar information such as those capture by n-gram statistics. We illustrate this using a bigram model. We use a sample based model of the bigram, instead of an histogram one. We represent the bigrams found in the sample set using a relationship matrix R:

$$R_i^j, \quad 1 \le i \le N, 1 \le j \le N \quad (17)$$

where we have

$$R_i^j = \begin{cases} 1, & \text{if } S_i \text{ can be the predecessor of } S_j \\ 0, & \text{if } S_i \text{ cannot be the predecessor of } S_j \end{cases} \quad (18)$$

R will be initialized according to a training text corpus. Entries are set to 1 or 0 if an example is either found or not found in the corpus. Note that this is different from histogram of counts used in traditional n-grams. To allow for me labels before and after each sign we use:

$$R_i^j = 1, \quad \text{if } i > V \text{ or } j > V \quad (19)$$

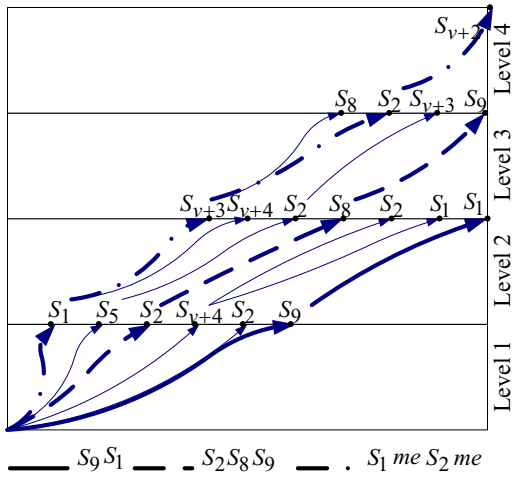


Figure 3. This figure illustrates the enhanced Level Building matching process. At levels 2-4 we obtained 3 matched sequences. The best one among these three will be returned as the matching result for these levels. Note all the signs S_{v+k} is actually a me segment with k frames, where we do not have any models associating with them.

After obtaining R, the eLB algorithm can be constrained with the predecessor relationship. Note that since we allow me label to exist between any two signs, a local backtracking may need to be performed while doing grammar checking. For example, assume at the current level we are examining the sign S_i. If the predecessor we found along the optimal path is a me label, we need to backtrack until we found the actual sign S_j along the optimal path. Grammar checking is performed finally between S_i and S_j.

We denote the result of the local backtracking for the minimum cumulative distance matrix A as:

$$B_l^i(m, k, j) = \beta \quad (20)$$

where S_β is the actual predecessor we found using the local backtracking scheme, when computing $A_l^i(m)$, along the path where the predecessor is $(l - 1, k, j)$.

Hence, to incorporate a grammar constraint into our system, we can update Eq. 12 and Eq. 15 as:

$$A_l^i(m) = \begin{cases} D(S_i, T_1^m), & \text{if } l = 1 \\ \min_{k,j} A_{l-1}^k(j) + D(S_i, T_{j+1}^m), & otherwise \\ R_\beta^i = 1, \beta = B_l^i(m, k, j) & \end{cases} \quad (21)$$

and

$$\psi_l^i(m) = \begin{cases} D(S_i, T_1^m), & \text{if } l = 1 \\ \arg\min_k A_{l-1}^k(j) + D(S_i, T_{j+1}^m), \\ R_\beta^i = 1, \beta = B_l^i(m,k,j) & otherwise \end{cases} \quad (22)$$

4. Choosing me Label Cost α

The choice of the cost for labeling a frame as me is a crucial one. We choose this by considering the distribution of match and non-match scores between signs in the training set. A match score is defined to the cost of matching different instances of the same sign and a non-match score is cost of matching instances of different signs. These scores are computed using dynamic warping and using the same frame to frame distance function used in the Level Building algorithm, (see $M(t^i, s_j^k)$ in Eq. 10). They are normalized by the length of the warping path. We then search of a threshold value that one can use to classify these scores into match and non-match ones. We choose the optimal α to be the optimal Bayesian decision boundary to accomplish this. However, instead of parametrically modeling each distribution (match and non-match) and then choosing the threshold, we empirically find the optimal value by sequentially searching for it. In essence, we are chooing the me labeling cost to be near the boundary of the match and non-match values.

5. Low-level Representation

Since the major contribution of this work is the enhanced Level Building algorithm, we just sketch the low-level representation used for completeness. Since our test is done based on pure video data, we developed a segmentation scheme to segment the hands out of the scene to form the feature vectors for each frame. This step is automatic, but has some noise.

The assumption that we make is that the hands move faster than other objects in the scene (including the face), and that the hand area can be somehwat localized by skin color detection. We used the mixed Gaussian model provided by Jones *et al.* [5], we use a safe threshold such that non skin pixels can be falsely classified as skin pixels.

We represent the possibly changing (but slowly) background, using a set of key frames. These key frames are identified as frames that are sufficiently different from each other. We sequentially search for them, starting from the first frame, which is always chosen to be a key frame. We compute the difference of any frame with previous key frame. If the non-component size in the thresholded difference image is large then the frame is labeled as the next key frame. This process continues until the end of the sequence. Then we compute the difference image of each frame to the key frames. The psuedocode of the approach enumerated below and some illustrative results are shown in Fig. 4.

For each sentence S_i with frames $< F_1, F_2, \cdots, F_N >$ repeat

1. Assign $k_1 = 1$,$m = 1$, $i = 2$. For frame $F_2, \cdots, F_N$ repeat
   (a) Compute difference image, D, between F_i and F_{k_m}. Find the largest connected component in D in terms of its number of valid pixels $Pixel_D$.
   (b) If $Pixel_D > threshold$, set $m = m + 1$, set $k_m = i$.
   (c) Set $i = i + 1$. If $i > N$ go to next step, else repeat above steps.
2. For each frame F_i, repeat
   (a) Compute a difference image SD, where $SD = (\Sigma_{j=1}^m abs(F_i, F_{k_j}))/(m-1)$
   (b) Mask SD with the skin likelihood image. Do edge detection on SD and obtain the edge image E.
   (c) Apply a dilation filter to E.
   (d) For each valid pixel in E, set the corresponding pixel of SD to be 0
   (e) Remove the small connected components in SD.
   (f) Extract the Boundary Image B.

Given the hand boundaries, we then capture the spatial structure by considering the distribution of the horizontal and vertical distances between pairs of pixels in it; we compute the joint relational histogram of the displacement between all pairs of coordinates on boundary images. We then represent these relational histograms, normalized to sum to one, as points in a space of probability functions (SoPF), like that used in [9]. The SoPF is constructed by performing principal component analysis of these relational histograms from the training set of images. The coordinates in the SoPF is the feature vector used in the matching process. We use the Mahalanobis distance as the distance measure.

6. Results

We have conducted extensive experimentation of the approach in the context of the task of recognizing *continuous* American Sign Language (ASL) sentences from image sequences. We present not only visual results of labeling continuous ASL sentences, but also quantify the performance. We compare the performance with that obtained by classical Level Building, which does not account for movement

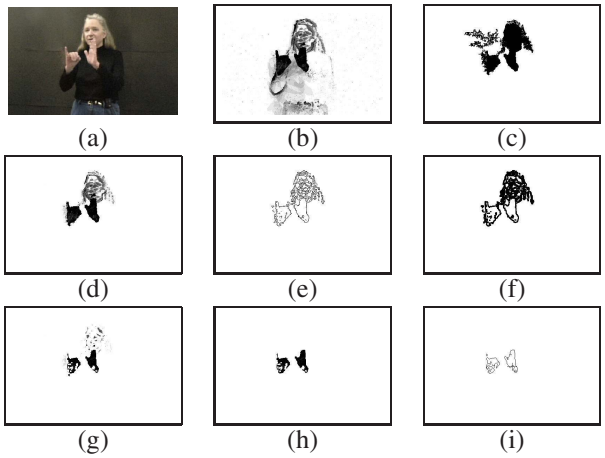


Figure 4. Intemediate results for the process of hand segmentation. (a) One frame in a sequence. (b) Consecutive frame difference image. (c) Skin pixels found. (d) Frame difference image with key frames (e) Edges found in (d). (f) After dialating (e). (g) After AND-ing the mask in (f) with (d). (h) After removing small components in (g). (i) Boundary of the component in (h). This the final hand candidate.

epenthesis. We were not able to compare with other explicit model based approaches to handling movement epenthesis since they require large training data, which, as far as we are aware of, is not available; we would need about 1000 labeled ASL sentences for the vocabulary size comparable to that used in this paper. In the results, we also present empirical evidence of the optimality of the choice of the α parameter is used to decide on the me mapping cost and present the impact of the grammar model.

6.1. Dataset

The vocabulary consists of signs that a deaf person would need to communicate with security personnel at airports. The video data is taken at 30 fps, with an image resolution of 460 by 290. Some frames were show in Fig. 1. There are 39 different signs that are articulated in 25 different continuous sentences. (Note that for approaches that explicitly model me we would need around 1000 sentences to capture the variations between signs.) Some signs appear more than once in some sentences. The total number of individual sign instances in the dataset is 73. There are 5 instances of each sentence. Some sentences have significant variations between multiple instances of the same sentence. This is introduced by signing the same sentences differently, for example, the English sentence 'if the plane is delayed, I'll be mad' can be signed as 'AIRPLANE POSTPONE AGAIN, MAD I' as well as 'AIRPLANE AGAIN POSTPONE, MAD I'. Since we have 5 samples per sentence, we perform 5-fold cross validation experiments. 4 of the folds are used for training and 1 for testing. This is repeated for the 5 possible choices of train/test partitions. The

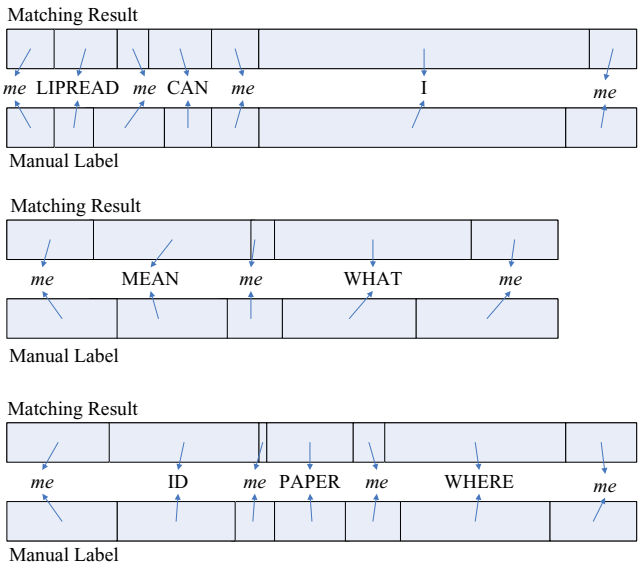


Figure 5. Diagrammatic representation of the labeling result for three sentences. Each horizontal bar represents a sentence that is partitioned into signs and me labels. The length of the horizontal bar is proportional to the number of frames in the sentence. For each sentence we present groundtruth partitioning and the algorithm output.

parameter alpha is trained on the training data (4 folds), so it is repeated for each of the 5 possible test experiments. The value of the alpha trained for each fold are 0.89, 0.85, 0.83, 0.83, 0.91. To enable us to quantify the performance, we manually labeled the frames corresponding to the signs in the sentences for the training partition. The grammar is trained based on a text corpus of 150 sentences that is independent of the video data.

6.2. Labeling Results

A labeling result for three sentences is diagrammatically presented in Fig 5. Each horizontal bar represents a sentence and is partitioned into signs or me blocks. The size of each block is proportional to the number of frames corresponding to that label. For each sentence we present the ground truth as determined by an ASL expert and the results from the algorithm. It is obvious that the signer is signing at different speeds for each sign. For instance, the sign I is spread over a large number of frames. The framework can easily handle such case. Apart from a 1 to 2 frame mismatch at the beginning and the end, the labeling match pretty well.

To quantitatively evaluate the results, we use errors as advocated in in [1]. If the recognized sentence inserted a sign that does not actually exist, one *insertion error* is spotted; if however the recognized sentence omitted a sign where it actually existed, one *deletion error* is counted; if the recognized sentence reports a wrong sign, we will consider it as a *substitution error*. We computed these errors

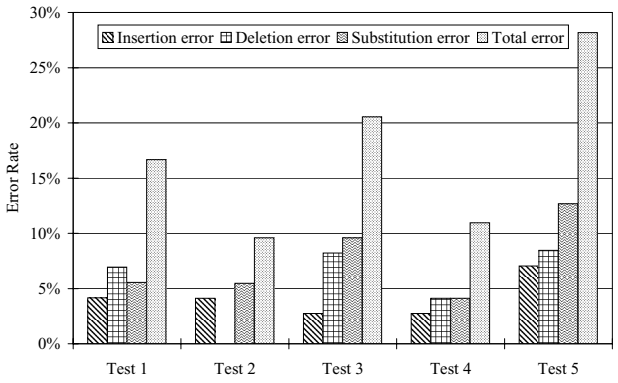


Figure 6. Sign level error rates, broken into insertion, deletion, and substitution, and for each test set in the 5-fold cross validation experiments with ASL data.

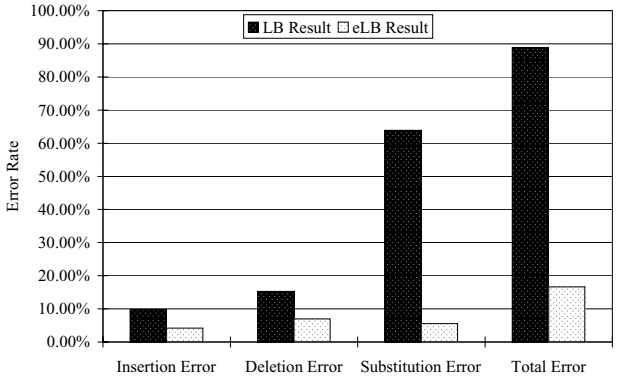


Figure 7. The error rates for enhanced Level Building, which accounts for movement epenthesis, and classical Level Building that does not account for movement epenthesis.

automatically by computing the Levenshtein distance using a dynamic programming approach [6] between the actual results and manually labeled ground truth.

Fig. 6 shows the error rates we obtained with the optimal α (more on this later) for each test set in the 5-fold validation experimentation, using a tri-gram model. The sign-level error rate for each test set ranges between 9% and 28%. On average, the error rate is 17%, with a corresponding correct recognition rate of 83%.

6.3. Classical vs. Enhanced

In Fig. 7 we present results of a head to head comparison of the error rates obtained using the enhanced Level Building algorithm presented here and classical Level Building that does not account for movement epenthesis. We found the insertion error has been decreased significantly by using the proposed method.

6.4. Grammar Model

Fig. 8 shows us side by side the error rates we obtained by using a tri-gram model and a bi-gram model. By using

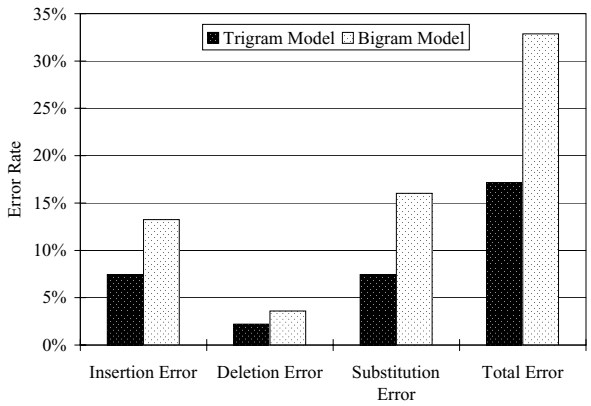


Figure 8. Error rates with trigram and bigram constraints.

tri-gram model, the average error rate dropped from 32% to 17%. The constraint imposed by a bi-gram model is more relaxed than imposed by a tri-gram model. It may be reiterated that we are using a 0-1 representation of the n-grams, i.e. for any instance of a relationship in the corpus the corresponding count is set to 1 otherwise it is zero.

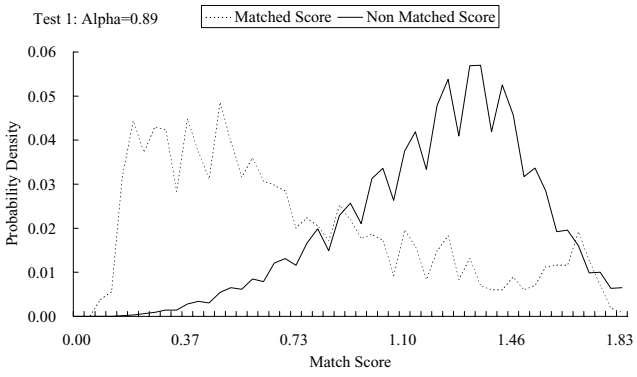
6.5. Parameter Choices

We assigned the parameters values as $L_{max} = 10$ and $N_{max} = 45$, which means we allow one sentence to have a maximum of 10 signs, and the maximum duration of movement epenthesis `me` to be 45 frames. We used the first 7 coefficients of the SOPF space representation as the feature vector. In our experiments, we have found these choices are quite stable. Varying them did not change the performance significantly. By far, the most important parameter is the `me` mapping cost α. As described in Section 4, we select the value of α is found to be the optimal Bayesian decision boundary between match and non-match scores. Fig. 9 (a) shows us the match and non-match scores on the training set for one of the 5-fold experiments. As we can see, a matched score usually average around 0.4, while a non-matching score is centered around 1.4. The optimal value for this training dataset is 0.89.

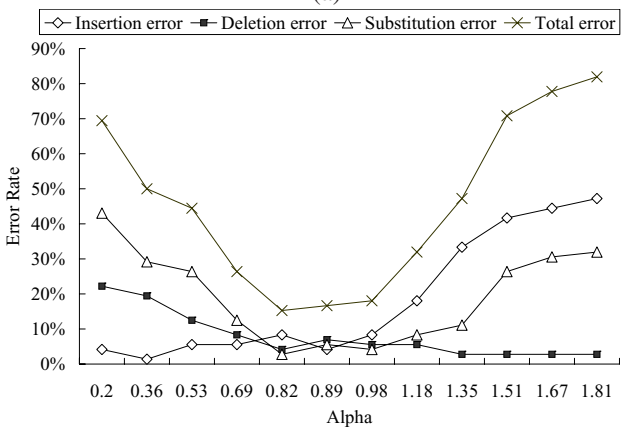
How good are the trained `me` labeling costs, α? To study this, we computed the best `alpha` that minimized the overall error rate on the test set. Fig. 9 (b) shows us the variation of the errors with different α for one of the test sets. We see that the automatically chosen α value of 0.89 is near the minimum of the actual error plots. In Table. 1 we list the errors with the automatically chosen αs for each of the 5-fold experiments and compare them with the actual possible minimums. The errors are within 4%. This shows that our method for choosing the optimal α is fairly robust.

7. Conclusions

We presented the enhanced Level Building algorithm, built around dynamic programming, to address the problem



(a)



(b)

Figure 9. Choosing the movement epenthesis (me) labeling cost, α. For one of the 5-fold experiments (a) shows us the match and non-match distance scores in the training set used to choose the optimal α. The optimal value is 0.89. (b) shows the variation of the errors with different choices of α.

Table 1. Error rates obtained with automatically (Opt.) chosen α and the one (Act.) that minimizes the error on the test set.

Test	Insertion		Deletion		Substitut.		Total	
	Opt.	Act.	Opt.	Act.	Opt.	Act.	Opt.	Act.
1	4%	8%	7%	4%	6%	3%	17%	15%
2	4%	0%	0%	3%	5%	5%	10%	8%
3	3%	1%	8%	5%	10%	10%	21%	16%
4	3%	3%	4%	4%	4%	4%	11%	11%
5	7%	3%	8%	1%	13%	13%	28%	17%
Avg.	7%	3%	2%	3%	7%	7%	17%	14%

of movement epenthesis in continous sign sentences. Our approach does not explicitly model movement epenthesis, hence the demand on annotated training video data is low. We compared the performance of enhanced Level Building with classical Level building algorithm, which has been proposed for connected word recognition in speech. We found significant improvements. Our extensive experimentation demonstrates the robustness of the matching process to different parameters. The developed enhanced Level Building algorithm solves the general problem of recognizing motion patterns from stream of compositions of motion patterns with portions, for which we do not have any model. Such situtation could arise in human computer interaction situation where one has to consider compositions of individual gestures or in long term monitoring of a person perform multiple activities.

The code and dataset used in this paper is available at *http://figment.csee.usf.edu/ ryang/CVAG/*.

8. Acknowledgment

This work was supported in part by the National Science Foundation under grant IIS 0312993.

References

[1] H. Brashear, T. Starner, P. Lukowicz, and H. Junker. Using multiple sensors for mobile sign language recognition. *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, pages 45–52, 2003.

[2] C.W. Sylvie and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):873–891, Jun 2005.

[3] W. Gao, G. Fang, D. Zhao, and Y. Chen. Transition movement models for large vocabulary continuous sign language recognition. In *FGR*, pages 553–558, 2004.

[4] G. T. Holt, P. Hendriks, and T. Andringa. Prospects and limitations of current automatic sign recognition research. *Sign Language Studies*, 6.4:416–437, 2006.

[5] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *Int. J. Comput. Vision*, 46(1):81–96, 2002.

[6] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, 1965.

[7] C. Myers and L. Rabiner. A level building dynamic time warping algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):284–297, 1981.

[8] L. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[9] I. Robledo and S. Sarkar. Representation of the evolution of feature relationship statistics: Human gait-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1323–1328, Oct 2003.

[10] C. Valli and C. Lucas. *Linguistics of American Sign Language: A Resource Text for ASL Users*. Gallaudet Univ. Press, 1992.

[11] C. Vogler and D. Metaxas. A framework of recognizing the simultaneous aspects of American Sign Language. *Computer Vision and Image Understanding*, 81(81):358–384, 2001.

[12] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *International Conference on Computer Vision*, page 1998, 363–369.

[13] R. Yang and S. Sarkar. Detecting coarticulation in sign language using conditional random fields. In *International Conference on Pattern Recognition*, pages 108–112, 2006.

[14] Q. Yuan, W. Gao, H. Yao, and C. Wang. Recognition of strong and weak connection models in continuous sign language. In *International Conference on Pattern Recognition*, page 10075, 2002.