Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming

Ruiduo Yang, Sudeep Sarkar, *Senior Member*, *IEEE*, and Barbara Loeding

Abstract—We consider two crucial problems in continuous sign language recognition from unaided video sequences. At the sentence level, we consider the movement epenthesis (me) problem and at the feature level, we consider the problem of hand segmentation and grouping. We construct a framework that can handle both of these problems based on an enhanced, nested version of the dynamic programming approach. To address movement epenthesis, a dynamic programming (DP) process employs a virtual me option that does not need explicit models. We call this the enhanced level building (eLB) algorithm. This formulation also allows the incorporation of grammar models. Nested within this eLB is another DP that handles the problem of selecting among multiple hand candidates. We demonstrate our ideas on four American Sign Language data sets with simple background, with the signer wearing short sleeves, with complex background, and across signers. We compared the performance with Conditional Random Fields (CRF) and Latent Dynamic-CRF-based approaches. The experiments show more than 40 percent improvement over CRF or LDCRF approaches in terms of the frame labeling rate. We show the flexibility of our approach when handling a changing context. We also find a 70 percent improvement in sign recognition rate over the unenhanced DP matching algorithm that does not accommodate the me effect.

Index Terms—Sign language, movement epenthesis, continuous gesture, segmentation, level building.

1 Introduction

Most approaches [1], [2] to continuous sign language recognition or continuous gesture recognition use hidden Markov models (HMM) [3] or dynamic time warping (DTW) [4], [5]. These matching processes were popularized by their effectiveness in speech recognition. HMM-based approaches are also popular for other types of sequences, such as text sequences [6]. Although a speech or text sequence can be considered to be similar to a sign language or gesture sequence in the sense that both of them can also be represented as a sequence of feature vectors, a video-based continuous sign language sequence does have vital differences. These differences make it hard to simply apply the successful approaches in speech recognition to sign language recognition.

One such differentiating aspect is the importance of movement epenthesis (me). During the phonological processes in sign language, sometimes a movement segment needs to be added between two consecutive signs to move the hands from the end of one sign to the beginning of the next [7]. This is called movement epenthesis (me) [1]. Fig. 1 shows an example of me frames. These frames do not correspond to any sign and can involve changes in hand shape, movement, and can be over many frames, sometimes equal in length of actual signs. Consequently, automated sign recognition systems need a way to ignore or identify and remove the me frames prior to translation of the true signs. The earliest work of which we are aware that explicitly modeled movement epenthesis in a continuous sign language recognition system with dedicated HMMs is by Vogler and Metaxas [8]. In another work [9], they also used context-dependent signs to model movement epenthesis and signs together. In a similar application of this approach, Yuan et al. [10] and Gao et al. [11] explicitly modeled movement epenthesis and matched with both sign and movement epenthesis models. The difference was that they used an automatic approach to precluster the movement epenthesis in the training data. More recently, Yang and Sarkar [12] used conditional random fields (CRF) to segment a sentence by removing me segments. However, this approach does not result in sign recognition, but just the segmentation of the sentence.

Although experimental results have shown that approaches that explicitly model movement epenthesis yield results superior to those ignoring movement epenthesis effects and context-dependent modeling [8], the question of scalability still remains. To obtain enough training data to model movement epenthesis is a real issue. With N signs, one may expect the number of movement epenthesis models to be N^2, i.e., quadratic in the number of signs. Also, to build movement epenthesis models, one has to label the associated frames in the training data, most likely manually and, hence,

- *R. Yang and S. Sarkar are with the Department of Computer Science and Engineering, 4202 E Fowler Ave., ENB 118, University of South Florida, Tampa, FL 33620. E-mail: {ryang, sarkar}@cse.usf.edu.*
- *B. Loeding is with the University of South Florida Polytechnic, 3433 Winter Lake Road, Lakeland, FL 33803. E-mail: bloeding@poly.usf.edu.*

Manuscript received 11 Jan. 2008; revised 10 July 2008; accepted 5 Jan. 2009; published online 16 Jan. 2009.
Recommended for acceptance by A. Martinez.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-01-0026.
Digital Object Identifier no. 10.1109/TPAMI.2009.26.

Fig. 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 that have no meaning and are known to be the me segment.

the model can be easily biased to this set of sentences. So, it is important during experimentation to separate the train and test data with respect to sentences as well, and not just with respect to instances of the same sentences.

Unlike previous approaches, we take a dynamic programming approach to address the problem of movement epenthesis, building upon the idea in [13]. Dynamic programming-based matching does not place demands on the training data as much as probabilistic models such as HMMs do. We illustrate the difference between our approach with the one that ignores movement epenthesis or the one that explicitly models movement epenthesis in Fig. 2. Fig. 2a represents a matching procedure that ignores me and matches all model signs in a model base to a test sentence. Note that the movement epenthesis between two signs can be falsely recognized as one of the signs. Fig. 2b, on the other hand, illustrates the process of explicitly modeling all the possible me frames, where the me frames in the test sequence are expected to be matched to the modeled me frames, not a sign. Fig. 2c sketches our approach. We have a model base that consists of all actual model signs, but not movement epenthesis. During the search for the optimal sign sequence in a sentence, we dynamically decide whether a match is a reliable match or not. If not, we label the test frame as a me. Determining the cost of this labeling is a crucial one and we have an effective, automated method for it. The entire process is embedded in a dynamic programming-based level building (eLB) algorithm coupled with a grammar model. The search process is conducted in a deterministic manner, where we use DTW, constrained by a grammar model. The advantage of the proposed matching process is that implicit segmentation of the sentence into signs happens without the need for modeling movement epenthesis. To create the model base, i.e., for training, we only need the sign frames in continuous sentences without the associated movement epenthesis.

Another crucial difference between speech and sign language recognition is that, while speech is sequential, sign language has both sequential and spatial aspects. Due to the sequential nature of speech and knowledge about ear physiology, it is somewhat easier to define features for speech, such as using frequency-based features [14], than for sign language. For video-based sign language sequence, a frequency domain representation of the frame cannot provide enough information for describing the local aspects such as hand shape, hand position, orientation, and motion. It is hard to detect spatially relevant parts in an image and to construct appropriate features. For instance, segmentation or tracking of the hands is hard even with a simple background due to the mutual occlusion of the hands, the changes in hand shape with motion, and the difficulty of localizing hands when in front of the face. Due to these complex low-level segmentation issues, previous continuous American Sign Language (ASL) recognition has mostly relied on assistive tools to obtain clean feature vectors. For example, Volger et al. [9], [15], [16] used a 3D tracking system and Cyber gloves, Wang et al. [17] used cyber gloves and 3D tracker, Starner et al. [18], [19] used color gloves, accelerometers, and head/shoulder mounted cameras, and Kadous [20] used power gloves. Although using assistive tools can yield better results, they also place added burden on the signer and can feel unnatural enough to even change the appearance of a normal sign. Some other approaches use only a single camera without assistive tools but with imaging constraints; for example, Bauer and Kraiss [21], [22] used a single color camera but with a uniform background and controlled clothing. Cui and Weng [23] used a segmentation scheme under a relatively complex background but their approach worked with image sequences with isolated signs. Other than this, Ding and Martinez [24] proposed methods to recover all of the

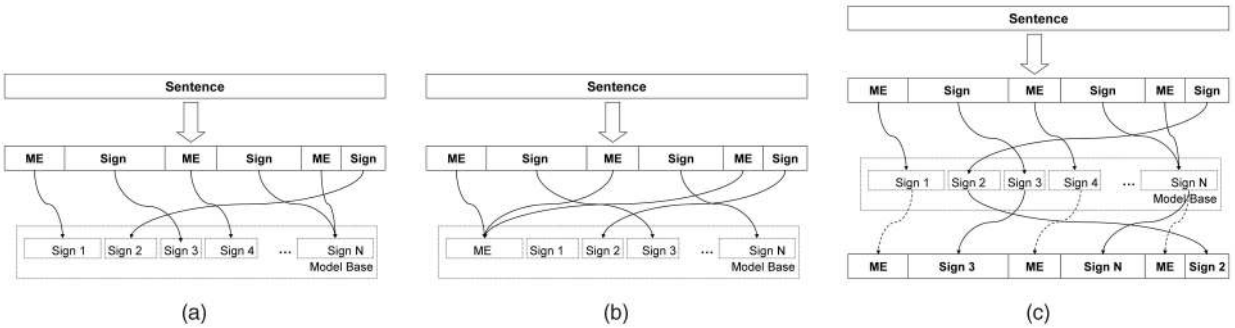

Fig. 2. Different approaches to handling movement epenthesis (me) in sentences: (a) If the effect of me is ignored while modeling, this will result in some me frames falsely classified as signs. (b) If me is explicitly modeled, building such models will be difficult when the vocabulary grows large. (c)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

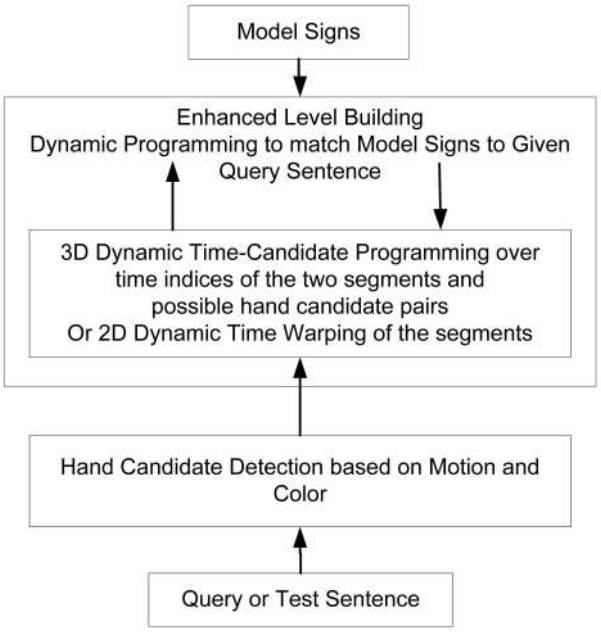

Fig. 3. Schematic of the approach. At the core are nested dynamic programming methods to simultaneously optimize over sign and movement epenthesis labels and multiple hand labels generated by low-level processing.

manual signals from a single unaided image sequence. Their approach is also based on simple background videos. In this work, we develop a theory to work with single camera video data without any assistive tools.

There are additional aspects that make the vision task harder. 1) We consider continuous sign language sentences as opposed to isolated signs; 2) we consider the issues that make hand segmentation hard, such as short sleeved shirts; and 3) we also consider complex background scenes. Our approach is able to cope with ambiguities in segmentation and the movement epenthesis problem in continuous sentences. The approach, depicted in Fig. 3, involves nested dynamic programming processes. Nested within the enhanced level building (eLB) dynamic programming method is another dynamic programming process that is concerned with matching single signs to segments of the observation sentence. We generate multiple hand candidates based on skin color and motion. Skeleton-based shape representation allows us to generate hand hypotheses for signers wearing both short sleeved and long sleeved shirts and also for the "hand across hand or face" cases. The hand candidates are paired to generate possible candidates for both of the hands. Then we link these pairs of hand candidates across the frames and match the sequence of candidates to the models using an enhanced version of the level building dynamic programming framework. At the sentence level of the enhanced level building process we match signs with sentences. During this matching, we allow for possible movement epenthesis labels based on matching scores. The combinatorics is constrained using a grammar model. The entire process simultaneously generates the final matching result and locates the hands.

From our literature survey, we noticed that there are previous approaches that have also used multiple hand candidate representations. For example, the combination of the top-down and bottom-up approach in gesture sequence recognition can be found in [25] and [26]. They both used skin and motion cues to generate multiple candidates. Yang and Sarkar [27], [28] used a grouping strategy to generate multiple candidates, including occlusion candidates, and fed them into an HMM framework for recognition of single signs. However, these previous works are all designed for isolated gesture recognition. Our multiple hand candidate approach is integrated into a level building framework to facilitate continuous sign language recognition. The elegant aspect of the approach is that we can handle me, hand localization and sentence level matching in the same formalism with one global optimization.

We experimented with four different kinds of single view video data sets. Some sample frames are shown in Fig. 4. The first data set has a simple background with a signer wearing a long sleeved shirt. We compare our methods with a traditional level building (LB) approach as well as the CRF and Latent Dynamic-CRF labeling approaches. The second data set is with a complex and changing background, and the third data set has simple background but with a short sleeved shirt, where we show improvement resulting from the use of the multiple candidate approach versus using the global feature approach. The last data set is part of the Purdue data set from [29], on which we test for across signer recognition.

In the following parts of the paper, we discuss the problem of me and the high-level DP process in Section 2. Section 3 describes the low-level DP process to handle the ambiguity problem. Section 4 presents the low-level processing and the generation of hand candidates. We then present the experiment results in Section 5 and our conclusions in Section 6.

2 Problem Formulation and High-Level Matching

We structure our notations as follows:

1. S_i: ith sign in a model base of size N, with V real signs and N_{max} virtual signs representing me labels of varying lengths from 1 to N_{max}. As indexed, the

Fig. 4. Example frames of the data sets; we denote them as D_1, D_2, D_3, and D_4 (Purdue data set).

first V models are for real signs, followed by the virtual me labels.
2. T: A test or query sentence of M frames, containing multiple signs.
3. $\mathbf{j}_l = \{(0), (1), (2), \ldots, (l)\}$: An ordered sequence of l integers. We will use this to represent the segmentation boundaries of a sentence, with the integers representing the ending frames of the segments.
4. $\mathbf{S}_l = \{S_{(1)}, S_{(2)}, \ldots, S_{(l)}\}$: A sequence of l sign labels. This allows us to distinguish between index of a sign in the model base and that in a particular label sequence.
5. L_{max}: Maximum number of signs in a sentence.
6. $T(i:j)$: Subsequence of T from frame i to frame j. Example usage include $T((i):(j))$, where (i) and (j) refer to entries in the ordered sequence or $S_{(k)}(i:j)$, referring to frames i through j in the kth sign, as listed in a label sequence.

A solution to the matching problem would consist of a segmentation of the sentence T into signs and movement epenthesis. Our objective is to find a sequence of sign and movement epenthesis (me) labels, $\mathbf{S}^*$, among all possible sign sequences such that the distance between $\mathbf{S}_l$ and T is minimized. That is,

$$\begin{aligned} R^* &= \arg\min_{l,\mathbf{j}_l,\mathbf{S}_l} A(\mathbf{S}_l, T) \\ &= \arg\min_{l,\mathbf{j}_l} \min_{\mathbf{S}_l} \sum_{i=1}^{l} D(S_{(i)}, T((i-1):(i))), \end{aligned} \tag{1}$$

where $D(.)$ is the function to compute the single sign matching cost with a segment of the test sequence. The nature of this cost function can differ based on the situation at hand. For instance, if we have good segmentation of hands and faces, then one could construct reliable feature vectors for each frame. In such situations, the distance would be constructed by dynamic time warping of the segments. If, on the other hand, we do not have reliable extraction of hands, then we suggest a more complex solution that involves optimizing over possible hand candidates. We will look into these distance computations methods, but, before that, let us consider how we perform the optimization in (1), given an appropriate distance measure.

2.1 The Enhanced Level Building Algorithm

The solution of (1) is over all of the possible *sign sequence* candidates, with all possible lengths for each sign. To control the combinatorics, we structure the search for the optimal solution using dynamic programming, specifically, the level building approach [14], and enhance it to allow for movement epenthesis me labels.

The overall minimization can be expressed recursively as optimization of one label and the minimum cost for the remaining sentence. If we structure this optimization separating the last label, we have

$$\begin{aligned} \min_{\mathbf{j}_l,\mathbf{S}_l} A(\mathbf{S}_l, T) = \min_{(l),S_{(l)}} \big(& D(S_{(l)}, T((l-1):(l))) \\ & + \min_{\mathbf{j}_{l-1},\mathbf{S}_{l-1}} A(\mathbf{S}_{(l-1)}, T((0):(l-1))) \big). \end{aligned} \tag{2}$$

Based on this decomposition of the problem, each level of the level building approach corresponds to the labels, in order, in the test sentence. Thus, the first level is concerned with the first possible label in the sentence. The first label could cover different possible lengths. The second level is concerned with the second possible label for the portion of the sentence that begins after the first label ends, 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.

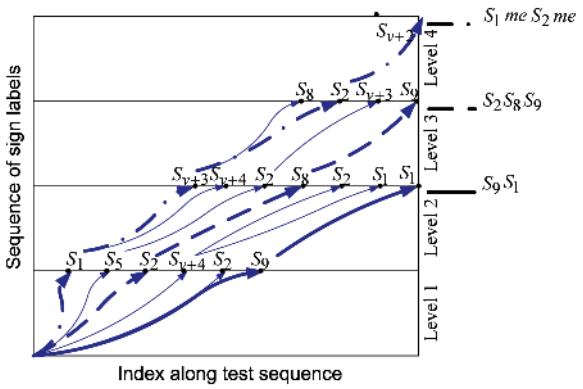

Fig. 5. The enhanced level building matching process. There are three complete sign label sequences, ending at levels 2, 3, and 4, respectively. The best one among these three will be returned as the matching result for these levels.

For each level l, we store the optimal cost for matching between sign S_i and with the ending frame as m using a three-dimensional cost array A of size $L_{max} \times N \times M$. The quantity $A(l,i,m)$ gives us the minimum cumulative score for matching l labels to the test sequence up to the mth frame, with the ith model sign, S_i, as the last label.

$$A(l,i,m) = \begin{cases} D(S_i, T(1:m)). & \text{if } l = 1, \\ \min\limits_{k,j} A(l-1,k,j) & \\ \quad + D(S_i, T(j+1:m)), & \text{otherwise.} \end{cases} \tag{3}$$

The optimal matching score D^* is $D^* = \min_{l,i} A(l,i,M)$.

To enable us to reconstruct the optimal sign sequence by backtracking, we use a predecessor array ψ, whose indices correspond to A: $\psi(l,i,m)$, $1 \le l \le L_{max}$,$1 \le i \le N$, $1 \le m \le M$, where

$$\psi(l,i,m) = \begin{cases} -1, & \text{if } l = 1, \\ \arg\min\limits_{k} A(l-1,k,j) & \\ \quad + D(S_i, T(j+1:m)), & \text{otherwise.} \end{cases} \tag{4}$$

Fig. 5 illustrates the possible matching sequences searched during the recursive search process. At the end of each level, we obtain the best matched sequences, for a portion of the test sequence. For example, level 1 is concerned with labeling the portion starting from the first frame. For each possible ending frame at a level, we obtained a best matching sign, for instance, $S_1, S_5, S_2, S_{v+4}, S_2, S_9$, shown in the figure. Then, at level 2, we again have a range of possible ending frames, with the starting frame after the ending frames from the first level. For each ending frame, we find the best cumulative matching score among all the signs and possible starting frame. We continue this process for all of

the levels. Matching that ends at the last frame results in a possible matching sequence. Three such complete sequences are shown in the figure: $\{S_9, S_1\}, \{S_2, S_8, S_9\}, \{S_1, \texttt{me}, S_2, \texttt{me}\}$. Note all of the label S_{V+k} is the me label over k frames.

The use of the me label is the essential difference between the classical level building formulation for recognizing connected words in speech and our formulation for recognition of connected signs in sign languages. We enhance the classical formulation by allowing for such a label, hence the name eLB. However, allowing for such a label is not equivalent to the addition of an additional sign label. It is not obvious how to choose the cost of me label as there are no real samples of it. One property could be that the cost be proportional to the length of me:

$$D(S_{V+k}, T(j+1, m)) = (m-j)\alpha. \tag{5}$$

This pushes the problem to choosing the proportionality constant, α, which is a penalty cost of assigning a me label to a frame. This penalty should be larger than a good match score we can find, since each time we find a good match to a portion of the unknown sequence from our database, we want to keep it. At the same time, the penalty should be smaller than a nonmatch score because, each time we cannot find any good match, we need to make sure the me match is selected. A nonmatch score is obtained when matching two different signs and a match score is obtained when matching different instances of the same sign. To estimate these scores we consider the distribution of match and nonmatch scores between signs in the training set, computed using dynamic time warping (discussed later). The overall distances are normalized by the length of the warping path. The distribution of these scores, typically has overlap. We search for a threshold value that one can use to classify these scores into match and nonmatch 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 nonmatch) and then choosing the threshold, we use a histogram-based representation to search for it.

2.2 Grammar Constraint

The explorations at each level can be constrained by grammar information such as those captured 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 and represent it using a relationship matrix $R(i,j)$, $1 \le i \le N$, $1 \le j \le N$, where we have

$$R(i,j) = \begin{cases} 1, & \text{if } S_i \text{ can be the predecessor of } S_j, \\ 0, & \text{if } S_i \text{ cannot be the predecessor of } S_j. \end{cases} \tag{6}$$

We set R based on observed instances in 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. Due to the limited nature of the samples, we do not use counts. Essentially, if we have some evidence, we set the probability of that occurrence as being one. This is a very liberal choice of grammar constraint. To allow for me labels before and after each sign we use $R(i,j) = 1$, if $i > V$ or $j > V$.

After obtaining R, the eLB algorithm can be constrained with the predecessor relationship based on the relationship matrix. Note that since we allow me label to exist between any two signs, a local backtracking may be needed while enforcing grammar checking. For example, assume at the current level we are deciding about the sign S_i. If the predecessor we found along the optimal path is a me label, we need to backtrack until we find a real sign S_p along the optimal path. Grammar checking is performed between S_i and S_p. Using the predecessor sign, S_p, found using local backtracking, we incorporate the grammar constraint into our system by modifying (3) and (4) as

$$A(l,i,m) = \begin{cases} D(S_i, T(1:m)), & \text{if } l = 1, \\ \infty, & \forall i \text{ s.t. } R(p,i) = 0, \\ \min\limits_{k,j} A(l-1,k,j) & \\ \quad + D(S_i, T(j+1:m)), & \text{otherwise}, \end{cases} \tag{7}$$

and

$$\psi(l,i,m) = \begin{cases} -1, & \text{if } l = 1, \\ -1, & \forall i \text{ s.t. } R(p,i) = 0, \\ \arg\min\limits_{k} A(l-1,k,j) & \\ \quad + D(S_i, T(j+1:m)), & \text{otherwise}. \end{cases} \tag{8}$$

Generalizations to n-gram statistics will involve an R function over $n-1$ predecessors and considerations of these predecessors in the above equation. When this number, n, is equal to the length of the sentences, we have sentence-based grammar, which is stronger than bigrams or trigrams. In the sentence-based grammar, any recognized sentence must be one sentence from a text corpus.

3 Single Sign Matching Cost

To compute the final optimal sequence using the eLB framework, we need to be able to compute the cost between a model sign with a subsequence of the test data, as $D(S_{(i)}, T((i-1):(i)))$ (1). There are two scenarios that we consider for this matching cost. First is when we have a single feature vector describing each image frame and any sign is a sequence of these feature vectors. This would be possible when one has fairly good segmentation, typically obtained by controlling the background and clothing. To compute the single sign matching cost under such situations, we can simply compute the DTW cost between the two sequences. As to the cost for matching one frame from a model to one observation frame, there are various choices possible, depending on the sophistication of the feature vector. We will discuss our particular choice in a later section.

The second scenario, which is the most common one, arises when we do not have good segmentation. This arises in uncontrolled imaging situations with complex background and lack of control over clothing. For each frame, we can have many possible hand candidate regions. Here the use of global features is obviously not reasonable. One has to allow for many possible hand candidates.

3.1 Paired Hand Candidates

Let the set of N_j hand blobs detected in frame j be: $\{p_1(j), p_2(j) \ldots, p_{N_j}(j)\}$. We consider all possible pairings of these primitives as candidates for the left and right hands, respectively.

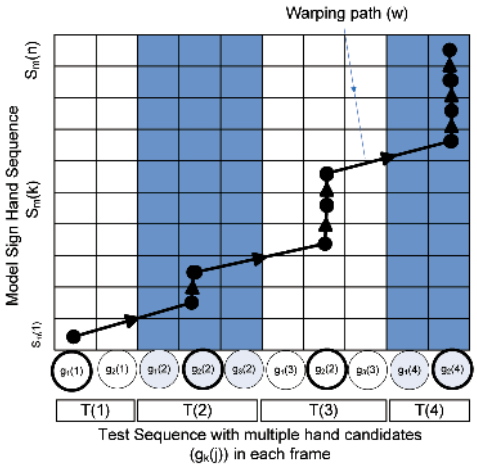

Fig. 6. Illustrating the 3D dynamic programming problem involved in matching a model sign sequence with a test subsequence. There are multiple candidates for each frame. The candidates for each frame selected by the illustrated warping path are represented by thick bordered circles. Note multiple model frames can match to a single test frame and vice versa.

$$G(j) = \{g_1(j) = \{p_1(j), p_2(j)\}, \ldots, \\ g_k(j) = \{p_{k_1}(j), p_{k_2}(j)\}, \ldots\}. \quad (9)$$

Note that we have $(N_j)^2 - N_j$ possible pairings. Each pairing is a possible observation of the hands. To match a model sign sequence to a test sequence, we have to not only match the frames, but also have to choose between the possible pairs in each frame.

3.2 Matching Hand Candidates

The goal of the matching of a model sign to a query subsequence is to find one candidate hand sequence that can be best mapped to the model sequence. In the model sign sequence S_m, we assume that we have hand labels. To represent each match, we denote $d(S_m(i), g_k(j))$ as the cost of matching the ith frame in S_m, with the kth hand-pair candidate from the jth frame in the test sequence, $g_k(j)$. For the experiments, we choose this matching cost to be the Mahalanobis distance between the feature descriptors, with a diagonal covariance matrix, calculated based on the model data sets. However, other choices are possible. The total distance will be the sum of these individual distances along possible matching curves, w. The solution will be the minimum value of this distance over all warping curves, w. An illustration of the 3D dynamic programming problem is shown in Fig. 6.

$$D(S_m, T(:)) = \min_w \left(\sum_{\{i,j,k\} \in w, g_k(j) \in T(j)} d(S_m(i), g_k(j))) \right). \quad (10)$$

3.3 Dynamic Programming Solution

We can minimize using dynamic programming. However, to contain the combinatorics, we limit the possible predecessors of a location (or node) on the warping curve. This is illustrated in Fig. 7 with some examples of allowed predecessors for a cell connected by arrows. Each cell in the lattice is indexed by the triple $\{i, j, k\}$, with the first two indices representing time along the model sign and test

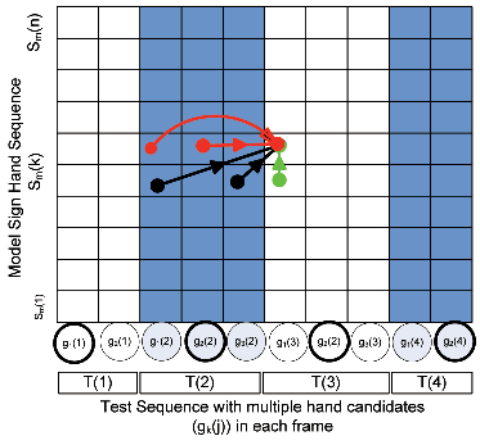

Fig. 7. Link constraints used to contain the combinatorics of dynamic programming. We allow for predecessors from up to one frame along the time axes and links from hand candidates constrained by the amount of frame-to-frame displacement.

sign, respectively, and the third index is for the possible hand pair in the jth test frame. For constraints along the time coordinates, we use the general, first order, local constraints [30]. For the hand candidate domain, we constrain the movement of possible hands by using a fairly liberal threshold, T_0, i.e., $m(g_k(j), g_r(j-1)) \leq T_0$. In the illustration in Fig. 7, we have a cell with five possible predecessors: one predecessor (in green) has the previous model frame, but the same test frame and hand candidates, two predecessors (red) have hand candidates from previous test frame but the same model frame, and two predecessors (black) have hand candidates from previous test frame and also the previous model frame.

The final dynamic programming update equations, incorporating the constraints, are as follows: Let $\mathrm{Cost}(i, j, k)$ represent the minimum cumulative cost of matching the model sequence, up to the ith frame, and up to kth hand candidates in the jth frame of the test sequence. We have the following recursive formula for dynamic programming:

$$\mathrm{Cost}(i, j, k) = d\left(S_m^i, g_k(j)\right) + \\ \min \begin{cases} \min\limits_{r, m(g_k(j), g_r(j-1)) \leq T_0} & \mathrm{Cost}(i, j-1, r), \\ \min\limits_{r, m(g_k(j), g_r(j-1)) \leq T_0} & \mathrm{Cost}(i-1, j-1, r), \\ \mathrm{Cost}(i-1, j, k). \end{cases} \quad (11)$$

The final solution is $D(S_m, T(:)) = \min_j \mathrm{Cost}(N_i, j, N_k)$.

4 Low-Level Representation

In this section, we describe our low-level processes that feed into the matching process. Many of the modules used are fairly standard ones, except for the background modeling scheme; therefore, we have placed this section after describing our core contributions, which is the matching process. To segment the hands automatically, we used skin color and motion. After segmenting the hands, we considered two kinds of feature vectors: a global feature vector and a part-based feature vector. We experimented with both these feature types in our experiments in head-to-head

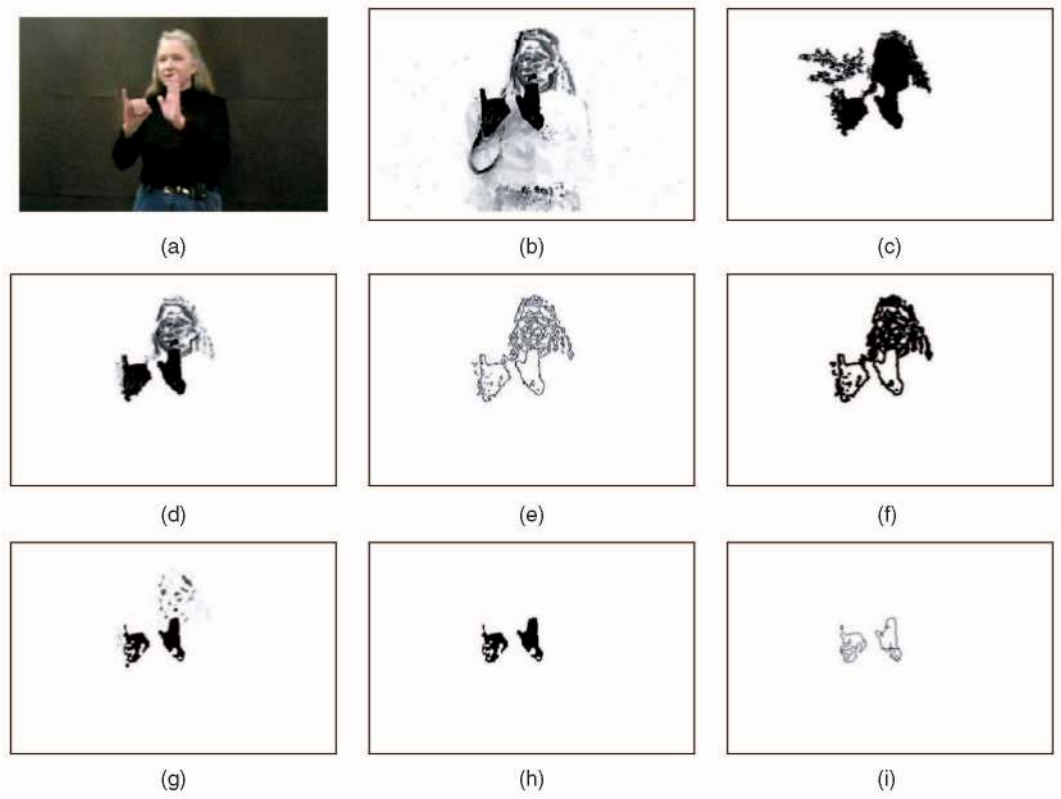

Fig. 8. Intermediate results from the hand segmentation process. (a) One frame in a sequence. (b) Consecutive-frame difference image. (c) Skin pixels found. (d) Frame difference image with keyframes. (e) Edges found in (d). (f) After dilating (e). (g) After AND-ing the mask in (f) with (d). (h) After removing small components in (g). (i) Boundary of the component in (h). This is the final hand candidate.

comparisons and also to demonstrate that the matching method outlined in this paper can be used in conjunction with different feature types.

4.1 Detection of Hands

Our assumption is that hands move faster than other objects in the scene (including the face) and that the hand area can be somewhat localized by skin color detection. We used the mixed Gaussian model of Jones and Rehg [31], with a safe threshold allowing for some amount of nonskin pixels to be falsely classified as skin pixels.

The specifics of the approach are outlined below and illustrated in Fig. 8. For each sentence T with N frames:

1. Assign first keyframe $k_1 = 1$, and initialize keyframe counter $m = 1$. For frame $i = 2, \ldots, N$:
   a. Compute difference image between $T(i)$ and $T(k_m)$. Find the largest connected component in the difference image in terms of its number of valid pixels N_p.
   b. If $N_p > T_1(threshold)$, set $m = m + 1$, set $k_m = i$.
   c. Set $i = i + 1$. If $i > N$ go to the next step, else repeat the above steps.
2. For frames $i = 1, \ldots, N$, repeat:
   a. Compute a difference image SD, where $SD = (\Sigma_{j=1}^{m} |S(i) - S(k_j)|)/(m-1)$.
   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. This step generates the motion-skin confidence map.
   f. Extract the Boundary Image B. Add a reference pixel (use the center of the frame or the center of the face) to B.

4.2 Global Features

We first generate the feature vectors using the boundary motion-skin confidence map obtained above (in step 2f). Given 2f, we capture the global spatial structure by considering the distribution (histogram) of the horizontal and vertical distances of each edge pixel and a reference point in the image. The reference point could be the image center or the centroid of the face region, detected by a face detector. We then represent these relational histograms, normalized to sum to one, as points in a space of probability functions (SoPF), like that used in [32]. The SoPF is constructed by performing a principal component analysis of these relational histograms from the model images. The coordinates in the SoPF is the feature vector used in the matching process. We use the Mahalanobis distance as the distance measure.

4.3 Multiple Hand Candidates

For cases with controlled background and clothing, as is the case with most sign language databases, the hand detection method outlined performs reasonably well. However, under

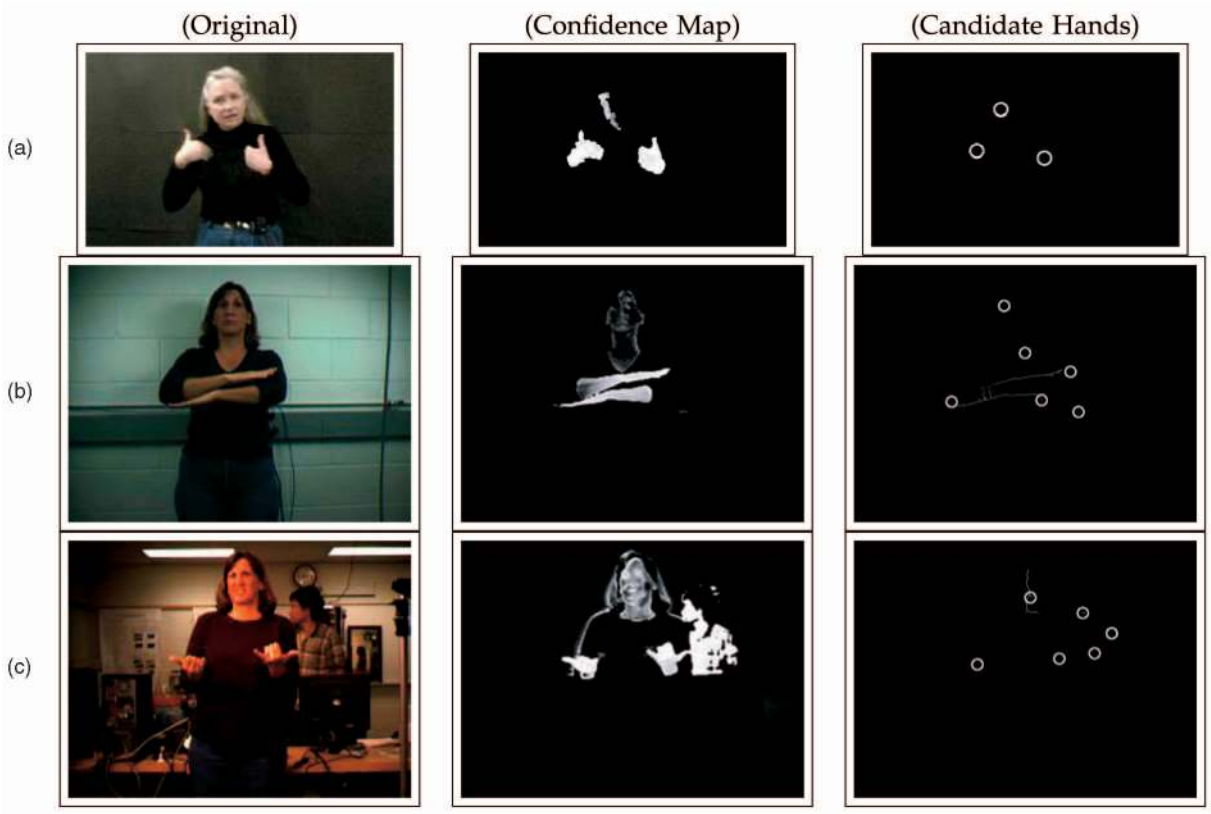

Fig. 9. Confidence map and the generated hand candidates, along with the medial axes for elongated components. There is movement in the background (c) and the signer is wearing short sleeved clothes (b).

uncontrolled cases where we can have nuisance motion-skin blobs in the background or if the signer is wearing a short sleeve shirt or even in the case where the signer's head (face) moves a lot, most hand detection algorithms, including ours, generate false alarms. To handle such cases, we generate multiple hand candidates and then select among them during the matching process as outlined earlier.

To construct the multiple candidates, we first represent the motion-skin confidence map as a collection of connected components. All connected components that are compact and small are selected to be hand candidates. The compactness is measured by dividing the number of pixels by the number of boundary pixels with a threshold T_2. The size is measured by the number of pixels with threshold T_3. The remaining components are too large to be the hand, but can arise from the merging of the arm with the hands. The hands in these cases would most likely be at one end of these merging shapes (for example, the hands or the fingers are most likely located at one end of the skin blobs, which can be represented by a leaf pixel of the media axis). To find these, we compute their medial axis by iteratively removing each boundary pixel that will not disconnect the connect component. Then we concentrate on all of the leaf pixels on the medial axis. These leaf pixels are then clustered using a nearest location neighbor clustering method with respect to a threshold T_4 until we get regions that are small enough to be hands. Fig. 9 shows some results for some sample frames in our three different data sets.

In our experiments, each hand primitive is described by its x coordinate, y coordinate, and average motion along the x and y directions on the image. This, we realize, handicaps our recognition process. Hand shape is an important component of sign recognition. There are signs with similar motion and location, but with different hand shapes. However, characterizing hand shape requires the precise segmentation of the hand boundary. This is especially hard for the case when hands and arm occlude each other or the face. Our approach to hand segmentation is not sufficient. We are also not aware of any algorithm that can precisely extract the shape of both hands, without the use of colored gloves, customized skin color modeling, or a manual initialization process. As hand segmentation research matures, hand shape features can be incorporated easily into our framework.

5 Results

We have conducted extensive experimentation with recognizing *continuous* ASL sentences from image sequences using our approach. We present not only visual results of labeling continuous ASL sentences, but also quantitative performance. We compared the performance with that obtained by classical level building, which does not account for movement epenthesis, and with frame labeling results obtained by two state-of-the-art methods: CRF [33] and Latent-Dynamic Conditional Random Field (LDCRF) [34].

We were not able to compare with other explicit model-based approaches to handle movement epenthesis and some generative methods, such as HMM, since they require large amount of training data, which we did not have. For the vocabulary size used in this paper, we would need about 1,000 labeled ASL sentences.

TABLE 1
Summary of the Three ASL Data Sets Used in This Paper

Name	D_1	D_2 + D_3	D_4
Resolution	460x290	640x480	640x480
# Training sentence	100	29 from $D2$ training	20
# Distinct Training Sentence	25	29	10
# Testing sentence	25	23 from $D2$ and 22 from $D3$	10
# Distinct Training Signs	40	39	99
# Two handed signs	21	23	53
# Same sentence in train and test?	Yes	No	Not exactly same
Background	Uniform	$D2$ is Complex with motion, $D3$ is static	Uniform
Short Sleeves	No	$D2$ is long sleeve, $D3$ is short sleeve	No
Number of Signers	1	1	3

All data sets are in color and 30 frames per second.

We also present empirical evidence of the optimality of the choice of the α parameter that is used to decide on the me mapping cost and present the impact of the grammar model on recognition.

5.1 Data Sets, Measures, and Experiments

Data sets. We have used four data sets, summarized in Table 1. Example frames from these four data sets are shown in Fig. 4. As we can see, the data sets vary in terms of the background. The background in data set D_1 is uniform, static with no texture. This is typical of sign language data sets. The background in D_3 is static but textured. The lighting in this data set is not directly on the subject. This data set is harder in terms of illumination and background conditions than D_1. This data set is not typical of sign language data sets, especially in the use of short sleeves. The data set D_2 is with complex background and moving people in the background. There are several patches in the background with skin color. For each frames in data sets D_2 and D_3 we have multiple hand candidates. Only for D_1 can we use global features. The fourth data set, D_4, is a subset of the Purdue Data Set [35], [29], where we have three signers signing 10 sentences. This was used for cross signer studies.

Train and test. The train and test for these data sets are structured as follows: In D_1, we have five samples per sentence. We performed fivefold cross-validation experiments, with four samples of each sentence for training and one for test. For D_2 and D_3, we have different sentences in the training and testing set. This is challenging for methods that explicitly or implicitly rely on me models. For experiments with data set D_4, we used two of the three signers as training and the third one for testing. This experiment was hard not only because it involved comparing across signers, but some of the semantically equivalent sentences differ in the sign ordering, i.e., the sequence of the signs are not the same.

Performance measures. To enable us to quantify the performance, we manually labeled the frames corresponding to the signs in the sentences. We also used the tool in [36] to manually select the true hand candidate in each model sign. To quantitatively evaluate the results, we used error measure, as advocated in [37]. We computed these errors automatically by computing the Levenshtein distance using a dynamic programming approach [38] between the found results and manually labeled ground truth. We refer to this error as "word-level rate." We also evaluated the framewise labeling result as the ratio of the total number of correctly labeled frames to the total number of frames. We call this the "frame-level rate."

Some words are in order to put the reported performances in context. While high recognition rates (on the order of $> 90\%$) of isolated ASL signs and isolated finger spelled signs have been reported, reported performances for recognition in continuous sentences vary quite a bit (58-90 percent [2]), depending on vocabulary sizes, length of sentences, and possibly other factors that have yet to be explored, such as the degree to which humans can recognize each sign under various conditions like complex background, etc.

TABLE 2
Outline of the Four Experimental Studies

Name	Study 1	Study 2	Study 3	Study 4
Purpose	eLB vs. LB, and α	eLB vs. CRF	Global vs. Hand candidates	Across signers
Datasets used	D_1	D_1	D_2 and D_3	D_4
Matching Algorithms	eLB and LB	eLB, CRF and LDCRF	eLB	eLB
Features	Global	Global	Hand candidates & Global	Hand candidates
Grammar	Bigram, Trigram	Trigram	Sentence	Sentence
Text corpus	Extended (150 sentences)	Extended (150 sentences)	Non-extended	Non-extended
Error	Word level rate	Frame level rate	Word level rate	Word level rate

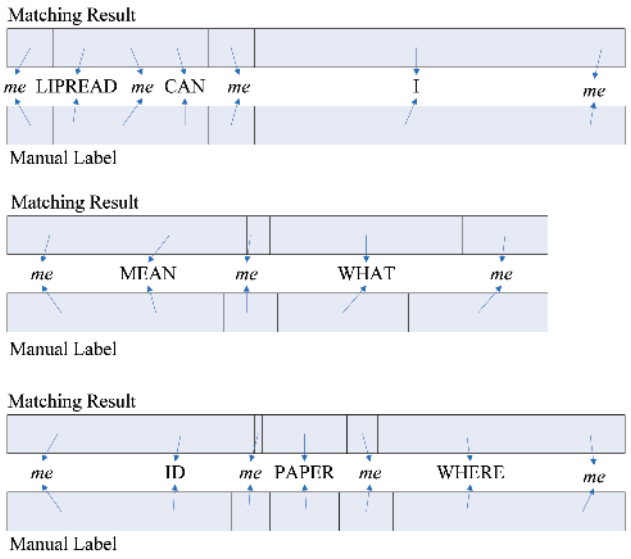

Fig. 10. Diagrammatic representation of the labeling result for three sentences. Each horizontal bar represents a sentence, partitioned into signs and me labels. The length of the horizontal bar is proportional to the number of frames in the sentence. For each sentence, we present ground truth partitioning and the algorithm output.

Studies. We conducted three studies. The details of the setup of the experiments are listed in Table 2. In the first study, we focused on the analysis of the eLB algorithm and the estimation of parameter α. We tested using both bigram and trigram grammar built using a text corpus of 150 sentences. The performance was measured using the word-level rate. In the second study, we compared our labeling approach with CRF/LDCRF approaches. For both study 1 and study 2, we used global feature vector, we also added reference pixels in the middle of the frame when we compute the relational distribution. Since CRF/LDCRF only produce a frame-level rate result, we used this as performance measure for this study. In the third study, we used D_2 and D_3 to test both global and part-based features under a changing context. In this study, we used a sentence-based grammar, which is stronger than just bigrams and trigrams. In the sentence-based grammar, any recognized sentence must be one sentence from the text corpus. For study 3, we used the face detection method from [39] to locate the face center and all positions were relative to this center.

Parameters. We used the same set of thresholds for all the experiments. We set these thresholds liberally based on the image size and the distance between the signer and the camera. Specifically, we used $T_0 = 100$ pixels, $T_1 = 300$ pixels, $T_2 = 2$, $T_3 = 4{,}000$ pixels, and $T_4 = \text{imageheight}/8$. For eLB setup, we assigned the parameters values as $L_{max} = 20$ and $N_{max} = 145$, which means we allow one sentence to have a maximum of 20 signs, and the maximum duration of movement epenthesis me to be 145 frames. We used the first seven coefficients of the SoPF space representation as the global feature vector [32]. In our experiments, we have found these choices to be quite stable. Varying them did not change the performance significantly.

5.2 Study 1: eLB versus LB with Grammar and Parameter Variation

The primary focus of the experiments in this study is to test the effectiveness of the eLB algorithm to overcome the me problem. We also study the choice of the me labeling cost α, the most crucial parameter. We conducted studies using data set D_1, where background related issues are least likely to confound the movement epenthesis recognition problem.

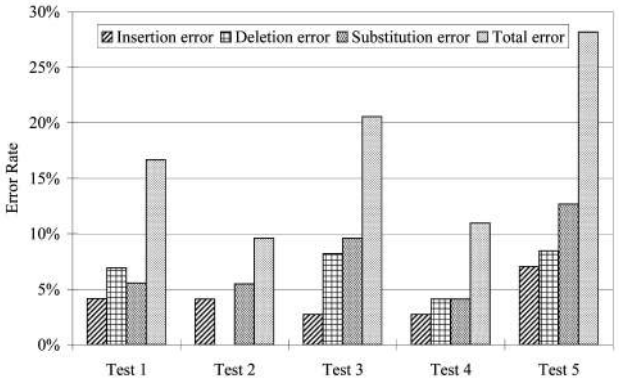

Fig. 11. Sign-level error rates using eLB on data set D_1 using global feature representation, broken into insertion, deletion, and substitution, and for each test set in the fivefold cross validation.

We show typical labeling results for three sentences in Fig. 10. Each horizontal bar represents a sentence that 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 handle such cases. Apart from a 1-2 frame mismatch at the beginning and the end, the labeling matches well.

Fig. 11 shows the sign-level error rates with the optimal α (more on this later) for each test set in the fivefold validation experimentation. This was using a trigram grammar model. The sign-level error rate for each test set ranges between 9 percent and 28 percent. On average, the error rate is 17 percent, with a corresponding correct recognition rate of 83 percent.

In Fig. 12, 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.

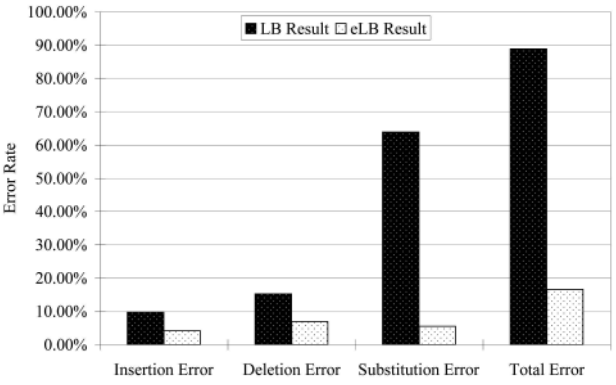

Fig. 12. The error rates for enhanced level building, which accounts for movement epenthesis, and classical level building, which does not account for movement epenthesis on data set D_1.

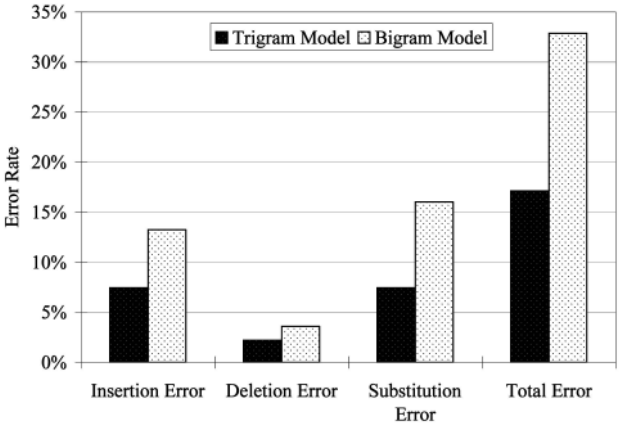

Fig. 13. Error rates with trigram and bigram constraints, which were constructed based on an ASL text corpus of 150 sentences.

We find that insertion errors have decreased significantly by using the proposed method.

Next, we studied the effect of the grammar model. Fig. 13 shows the error rates we obtained by using a trigram model and a bigram model. We constructed the grammar models based on a text corpus of 150 sentences. These sentences did not all have corresponding video data. By using the trigram model, the average error rate dropped from 32 percent to 17 percent. The constraint imposed by a bigram model is more relaxed than that imposed by a trigram 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 0.

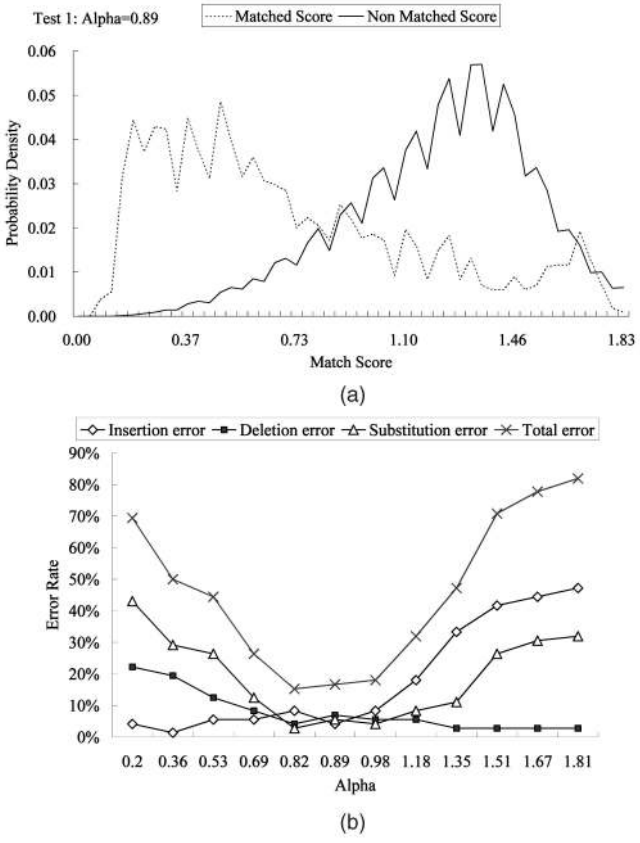

Fig. 14. Experiments with the movement epenthesis (me) labeling cost, α. (a) The match and nonmatch distance scores in the training set used to choose the optimal α for one of the fivefold experiments. The optimal value is 0.89. (b) The variation of the errors with different choices of α.

TABLE 3
Error Rates with eLB on Data Set D_1, with Automatically (Auto) Chosen α and the One (Opt.) that Minimizes the Error on Each Test Set

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

By far, the most important parameter is the me labeling cost α. As described earlier, we select the value of α to be the optimal Bayesian decision boundary between match and nonmatch scores. Fig. 14a shows the match and nonmatch scores on the training set in data set D_1 for one of the fivefold experiments. As we can see, a matched score usually average around 0.4, while a nonmatching score is centered around 1.4. The optimal value for this training data set is 0.89.

How good are the trained me labeling costs, α? To study this, we computed the best α that minimized the overall error rate on the test set. Fig. 14b shows 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 error plots. In Table 3, we list the errors with the automatically chosen αs for each of the fivefold experiments and compare them with the actual possible minimums. The errors are within 4 percent. This shows that our method for choosing the optimal α is fairly robust.

5.3 Study 2: Comparison with Other Approaches

We compare the performance of our approach with two state-of-the-art methods: CRF [33] and LDCRF [34]. We use the code from [34] to generate our results. These particular models have been developed in gesture recognition context, where the labels correspond to gestures. The posterior probability is maximized or estimated directly during training and testing. For both methods, we used a chaining structure, with three hidden states for each label in the LDCRF. In Table 4, we quantify performance using the frame-level error rate, i.e., what percentage of the frames are wrongly classified in the test set. As we can see, CRF

TABLE 4
Framewise Labeling Performance of eLB, LB, CRF, and LDCRF

Methods	eLB	LB	CRF	LDCRF
Parameters	1	0	1968	15990
Classes	41	40	41	41
Dataset used	D_1	D_1	D_1	D_1
Grammar Model	Trigram	Trigram	N/A	N/A
Total Test Frames	2234	2234	2234	2234
Correct Labeled Frames	1530	406	642	460
Error rate	31%	82%	71%	89%

and LDCRF perform quite poorly. Although CRF [33] and LDCRF [34] have shown improved results for limited number of labels, in our experiments we had to use them for $40+$ labels. As the number of possible labels increases, the number of parameters that need to be estimated increases significantly for these models. This makes the training starved for data. Also, both CRF and LDCRF implicitly model me as one single class, which is not a realistic model.

5.4 Study 3: Global Features versus Multiple Local Candidates

The eLB framework can handle both global features that are computed based on the whole image frame and local features, computed for hand candidates. In this study, we show two advantages of the nested framework with these feature types. One of the advantages of the framework is that the training is solely based on the sign model. We do not have to retrain the models or insert new me training data when the conversation context changes as long as the sign vocabulary is the same. One example where this flexibility will be useful is for short question-answer format communication between the computer and a Deaf person. Each time the computer asks a question, it can use the context of the conversation to anticipate the possible answers (sentences) from the Deaf person. Thus, the context is dynamic. For each question, the possible answer set is different, but small. However, we do not have to retrain for each context. We only need to dynamically change the text corpus used to model the grammar context, which is easy to accomplish. The other advantage of the framework is that we handle not only the movement epenthesis, but also segmentation issues related to complex backgrounds, short sleeves, and hands occlusions. We do not have to make definitive decisions about the hand positions during low-level segmentation.

In the first set of experiments in this study, we show the ability of the eLB framework to accommodate a dynamically changing conversation context. We used a portion of D_2 as the training data, the other portion of D_2, along with D_3, was the testing data, comprised of 45 sentences. We randomly picked 5 or 20 sentences from these 45 sentences and performed recognition. We repeated the process 10 times. Each time the context was changed, we only switched the text corpus from which the grammar constraints were derived.

Fig. 15 shows the result of the tests. Each bar of Fig. 15 shows one recognition result based on one set of randomly picked test sentences and its corresponding text corpus. From the results, we can observe that context is important in recognizing the sentences. With five sentences in the text corpus, we can achieve very good recognition result. We can see that 7 of the 10 randomly conducted tests has a error rate of 0 percent. However, when we increase the number of sentences in the context, the error rate increases to above 30 percent.

The results depend on the data set and features used. In the previous study, with simple background data, we had obtained a recognition rate of 83 percent. Here, with a relatively uncontrolled background, performance falls as the number of sentences to recognize increases. Features are also important. From the figure, we can see that the global feature is not a reasonable choice for this data set because of the complex and moving background or short sleeve clothes

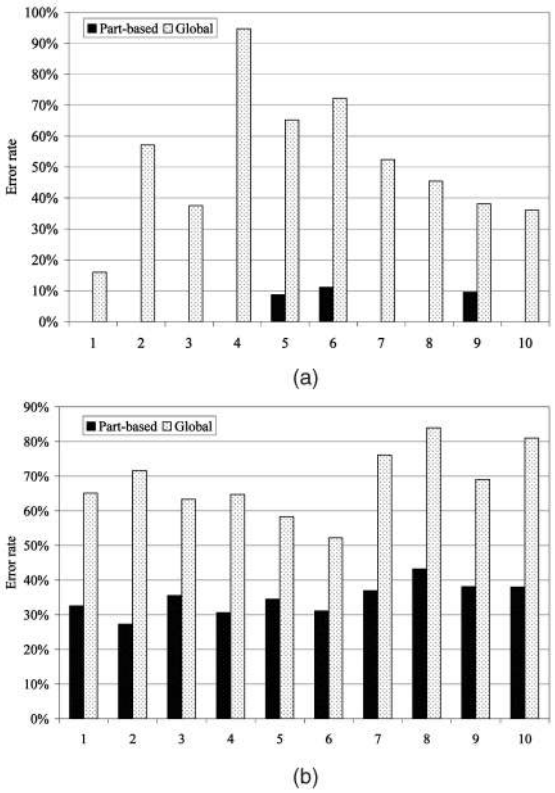

Fig. 15. Total error rate with changing test contexts, with part and global features, with a context of (a) 5 test sequences and (b) 20 test sequences. For each scenario, we present results from 10 sets of randomly picked test sequences from a database of 45 sentences. Note that for (a) the error rates were zero in some cases for the part-based features.

used. For local features, our choice of just location and motion of hands also limits the discriminative abilities. Future possibilities exist for increasing performance by using richer features incorporating hand shapes and facial expressions in this framework.

Figs. 16 and 17 show examples of the hand candidates selected by the eLB algorithm, for two continuous sentences, one with moving background and one with short sleeve clothes, respectively. On the left of each image block, we have the detected hands shown as red circles overlaid on the original image, and all of the candidate hands for the corresponding frame are shown on the right. Note that there are no selected hand candidates (on the left side) for frames labeled as me. It is also interesting to see that, for the sentence in Fig. 16, although the sentence recognition is correct (which is what we want), the framewise labeling is not completely right. This is due to the fact that we only use very coarse features, such as position and moving directions, to conduct the match, the signs in between can be easily mixed up with each other. However, the eLB framework can still make the final recognition for the sentence correct based on context of the text corpus.

5.5 Study 4: Across Signer Recognition

In this study, we focused on testing across signer recognition using the proposed framework. For multiple signer data, in addition to the expected variations related to speed, motion amplitude, etc., there also exists a larger source of variation. Signers might sign a sentence in different sign orders. For

Fig. 16. The labeling results for the sequence "FINISH BUY TICKET NOW FINISH." On the left of each image block we have the detected hands shown as red circles overlaid on the original image, and all of the candidate hands for the corresponding frame are shown on the right. The frames are arranged in row-scan order, left to right and then down. The video corrresponding to this figure is available as supplemental material which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2009.26.

example, signer A may sign "READ NEWSPAPER I...," while signer B may sign "NEWSPAPER READ I ..." This will make the explicit training of me harder. However, our proposed framework does not rely on explicitly training of me. Hence, we expect it to work for these cases.

We conducted this experiment using a subset of the Purdue data set [35], [29], spanning three signers. Each of the signers signs 10 sentences once. For each sentence, the three signers are free to communicate using different sign orders or even choose from multiple possibilities for some signs. We conducted three tests, one for each of the three signers, using the other two as training. We had two different context setups for each test, using 5 and all 10 sentences. In Fig. 18, we show the overall error rates. The recognition rate tops at

Fig. 17. The labeling results for the sequence "MY TICKET HERE." On the left of each image block we have the detected hands shown as red circles overlaid on the original image, and all of the candidate hands for the corresponding frame are shown on the right. The frames are arranged in row-scan order, left to right and then down. The video corrresponding to this figure is available as supplemental material which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2009.26.

80 percent. We do see a drop in performance over intersigner recognition. Given the hard conditions, even for across signer tests, this is not surprising. As expected for smaller sentence context, performance is better than for larger contexts.

6 Conclusions

We designed and explored the enhanced level building algorithm, built around dynamic programming, to address the problem of movement epenthesis in continuous sign sentences. Our approach does not explicitly model movement epenthesis, hence the demand on annotated training

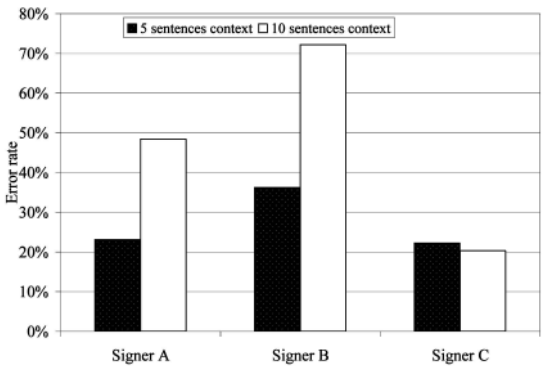

Fig. 18. Across signer recognition on data from three signers in the Purdue sign language data set. There were three experiments, with two signers in the train set and the third signer as test. For each experiments, we had two different context setups by using 5 and 10 sentence contexts.

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. To overcome the low-level hand segmentation errors, we incorporated another dynamic programming process, nested within the first one, to optimize over possible choices from multiple hand candidates. Our results showed that the part-based candidate approach works better under moving background and short sleeve situations. Our extensive experimentation also demonstrates the robustness of the matching process with different parameters. In the context of ASL, this work advances recognition of signs in sentences, while accounting for movement epenthesis, and we also contribute toward the ability to handle general backgrounds and relaxation of clothing restrictions. The developed enhanced level building algorithm solves the general problem of recognizing motion patterns from a stream of compositions of motion patterns with intervening portions, for which we do not have any model. Such situations could also arise in general human computer interaction situations where one has to consider compositions of individual gestures or in long-term monitoring of a person performing multiple activities.

ACKNOWLEDGMENTS

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

REFERENCES

[1] C. Sylvie and S. Ranganath, "Automatic Sign Language Analysis: A Survey and the Future Beyond Lexical Meaning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 6, pp. 873-891, June 2005.

[2] B. Loeding, S. Sarkar, A. Parashar, and A. Karshmer, "Progress in Automated Computer Recognition of Sign Language," *Lecture Notes in Computer Science,* vol. 3118, pp. 1079-1087, Springer, 2004.

[3] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, no. 2, pp. 257-286, Feb. 1989.

[4] C. Myers and L. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. 29, no. 2, pp. 284-297, Apr. 1981.

[5] J. Lichtenauer, E. Hendriks, and M. Reinders, "Sign Language Recognition by Combining Statistical DTW and Independent Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 11, pp. 2040-2046, Nov. 2008.

[6] M. Skounakis, M. Craven, and S. Ray, "Hierarchical Hidden Markov Models for Information Extraction," *Proc. Int'l Joint Conf. Artificial Intelligence,* 2003.

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

[8] C. Vogler and D. Metaxas, "A Framework of Recognizing the Simultaneous Aspects of American Sign Language," *Computer Vision and Image Understanding,* vol. 81, no. 81, pp. 358-384, 2001.

[9] C. Vogler and D. Metaxas, "ASL Recognition Based on a Coupling between HMMs and 3D Motion Analysis," *Proc. Int'l Conf. Computer Vision,* pp. 363-369, 1998.

[10] Q. Yuan, W. Gao, H. Yao, and C. Wang, "Recognition of Strong and Weak Connection Models in Continuous Sign Language," *Proc. Int'l Conf. Pattern Recognition,* vol. 1, pp. 75-78, 2002.

[11] W. Gao, G. Fang, D. Zhao, and Y. Chen, "Transition Movement Models for Large Vocabulary Continuous Sign Language Recognition," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition,* pp. 553-558, 2004.

[12] R. Yang and S. Sarkar, "Detecting Coarticulation in Sign Language Using Conditional Random Fields," *Proc. Int'l Conf. Pattern Recognition,* pp. 108-112, 2006.

[13] R. Yang, S. Sarkar, and B.L. Loeding, "Enhanced Level Building Algorithm for the Movement Epenthesis Problem in Sign Language Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2007.

[14] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition.* PTR Prentice Hall, 1993.

[15] C. Vogler and D. Metaxas, "Handshapes and Movements: Multiple-Channel ASL Recognition," *Lecture Notes in Artificial Intelligence,* vol. 2915, pp. 247-258, Springer, 2004.

[16] C. Vogler, H. Sun, and D. Metaxas, "A Framework for Motion Recognition with Application to American Sign Language and Gait Recognition," *Proc. Workshop Human Motion,* pp. 33-38, 2000.

[17] C. Wang, W. Gao, and S. Shan, "An Approach Based on Phonemes to Large Vocabulary Chinese Sign Language Recognition," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition,* pp. 393-398, 2002.

[18] H. Brashear, V. Henderson, K.-H. Park, H. Hamilton, S. Lee, and T. Starner, "American Sign Language Recognition in Game Development for Deaf Children," *Proc. Int'l ACM SIGACCESS Conf. Computers and Accessibility,* pp. 79-86, 2006.

[19] T. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition,* pp. 189-194, 1995.

[20] M. Kadous, "Machine Translation of AUSLAN Signs Using Powergloves: Towards Large Lexicon-Recognition of Sign Language," *Proc. Workshop the Integration of Gesture in Language and Speech,* pp. 165-174, 1996.

[21] B. Bauer, H. Hienz, and K.-F. Kraiss, "Video-Based Continuous Sign Language Recognition Using Statistical Methods," *Proc. Int'l Conf. Pattern Recognition,* vol. 2, pp. 2463-2466, 2000.

[22] B. Bauer and K.-F. Kraiss, "Video-Based Sign Recognition Using Self-Organizing Subunits," *Proc. Int'l Conf. Pattern Recognition,* vol. 2, pp. 434-437, 2002.

[23] Y. Cui and J. Weng, "Appearance-Based Hand Sign Recognition from Intensity Image Sequences," *Computer Vision and Image Understanding,* vol. 78, no. 2, pp. 157-176, 2000.

[24] L. Ding and A. Martinez, "Recovering the Linguistic Components of the Manual Signs in American Sign Language," *Proc. IEEE Conf. Advanced Video and Signal-Based Surveillance,* 2007.

[25] Y. Sato and T. Kobayashi, "Extension of Hidden Markov Models to Deal with Multiple Candidates of Observations and Its Application to Mobile-Robot-Oriented Gesture Recognition," *Proc. Int'l Conf. Pattern Recognition,* pp. 515-519, 2002.

[26] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "Simultaneous Localization and Recognition of Dynamic Hand Gestures," *Proc. IEEE Workshop Motion and Video Computing,* vol. 2, pp. 254-260, 2005.

[27] R. Yang and S. Sarkar, "Gesture Recognition Using Hidden Markov Model from Fragmented Observations," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 766-773, 2006.

[28] R. Yang and S. Sarkar, "Coupled Grouping and Matching for Sign and Gesture Recognition," *J. Computer Vision and Image Understanding*, vol. 113, no. 6, pp. 663-681, 2009.
[29] R. Wilbur and A. Kak, "Purdue RVL-SLLL American Sign Language Database," Technical Report 06-12, School of Electrical and Computer Eng., Purdue Univ., http://RVL.ecn.purdue.edu/database/Wilbur_Kak.html, 2006.
[30] H. Silverman and D. Morgan, "The Application of Dynamic Programming to Connected Speech Recognition," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 26, no. 6, pp. 575-582, July 1990.
[31] M. Jones and J. Rehg, "Statistical Color Models with Application to Skin Detection," *Int'l J. Computer Vision*, vol. 46, no. 1, pp. 81-96, 2002.
[32] I. Robledo and S. Sarkar, "Statistical Motion Model Based on the Change of Feature Relationships: Human Gait-Based Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1323-1328, Oct. 2003.
[33] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Conditional Models for Contextual Human Motion Recognition," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 210-220, 2006.
[34] L. Morency, A. Quattoni, and T. Darrell, "Latent-Dynamic Discriminative Models for Continuous Gesture Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2007.
[35] A.M. Martinez, R.R. Wilbur, R. Shay, and A. Kak, "Purdue RVL-SLLL ASL Database for Automatic Recognition of American Sign Language," *Proc. Int'l Conf. Multimodal Interfaces*, 2002.
[36] R. Yang, S. Sarkar, B.L. Loeding, and A.I. Karshmer, "Efficient Generation of Large Amount of Training Data for Sign Language Recognition: A Semi-Automatic Tool," *Proc. Conf. Computers Helping People with Special Needs*, pp. 635-642, 2006.
[37] H. Brashear, T. Starner, P. Lukowicz, and H. Junker, "Using Multiple Sensors for Mobile Sign Language Recognition," *Proc. IEEE Int'l Symp. Wearable Computers*, pp. 45-52, 2003.
[38] V. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Doklady Akademii Nauk SSSR*, vol. 163, no. 4, pp. 845-848, 1965.
[39] *Open Source Computer Vision Library*, http://www.intel.com/research/mrl/research/opencv, 2008.

Ruiduo Yang received the bachelor of science degree in computer science from Peking University, Beijing, China, in 2001, and the master of philosophy degree in computer science and engineering from Hong Kong University of Science and Technology, China, in 2003. He also received the PhD degree from the Department of Computer Science and Engineering in the University of South Florida, Tampa, in 2008. His research interests include sign language and gesture recognition, machine learning, sequence recognition, video analysis, and video coding. He has coauthored more than 10 publications in the field of pattern recognition and video processing. He has served as a reviewer for the *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, the *Journal of Pattern Recognition*, and the *IEEE Transactions on Biomedical Engineering*.

Sudeep Sarkar received the BTech degree in electrical engineering from the Indian Institute of Technology, Kanpur, in 1988. He received the MS and PhD degrees in electrical engineering, on a University Presidential Fellowship, from The Ohio State University, Columbus, in 1990 and 1993, respectively. Since 1993, he has been with the Computer Science and Engineering Department at the University of South Florida (USF), Tampa, where he is currently a professor. His research interests include perceptual organization in single images and multiple image sequences, automated sign language recognition, biometrics, and nanocomputing. He is the coauthor of the book *Computing Perceptual Organization in Computer Vision* (World Scientific). He is also the coeditor of the book *Perceptual Organization for Artificial Vision Systems* (Kluwer Academic Publishers). He was a recipient of the US National Science Foundation CAREER award in 1994, the USF Teaching Incentive Program Award for undergraduate teaching excellence in 1997, the Outstanding Undergraduate Teaching Award in 1998, and the Theodore and Venette Askounes-Ashford Distinguished Scholar Award in 2004. He served on the editorial boards for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1999-2003) and *Pattern Analysis & Applications Journal* (2000-2001). He is currently serving on the editorial board of *Pattern Recognition Journal*, the *IEEE Transactions on Systems, Man, and Cybernetics, Part-B*, *Image and Vision Computing*, and *IET Computer Vision*. He is a fellow of the IAPR, a senior member of the IEEE, and a member of the IEEE Computer Society.

Barbara Loeding received a double BS degree in communication disorders and psychology from the University of Minnesota in 1975. She received the MS degree in speech language pathology in 1979 (Minnesota State University Mankato) and the PhD degree in special education, while on a fellowship, from Purdue University, Indiana, in 1989. Since 1989, she has been with the Department of Communication Sciences and Disorders (Tampa) and then with the Department of Special Education at the University of South Florida, Lakeland, where she is currently an associate professor. Her research interests include accessibility of technology for people with disabilities, improvement of communication between hearing and deaf individuals through the use of automated American Sign Language recognition and automated American Sign Language synthesis systems, improvement of human-computer interfaces, usability testing, and functional measures of performance evaluation of vision systems. She is the recipient of the Johns Hopkins Award for the development of an innovative computerized program to assess interpersonal skills of people who use sign language. She has collaborated with DePaul University's ASL Synthesizer Project and prior to that with Hill Abraham for the development of a video-based English to Sign translation program. She has served on several editorial boards for the field of speech language pathology, including as associate editor for the International Society for Augmentative and Alternative Communication, has presented internationally, and has published numerous articles on her work.