Sign Language Spotting with a Threshold Model Based on Conditional Random Fields

Hee-Deok Yang, *Member*, *IEEE*, Stan Sclaroff, *Senior Member*, *IEEE*, and Seong-Whan Lee, *Senior Member*, *IEEE*

Abstract—Sign language spotting is the task of detecting and recognizing signs in a signed utterance, in a set vocabulary. The difficulty of sign language spotting is that instances of signs vary in both motion and appearance. Moreover, signs appear within a continuous gesture stream, interspersed with transitional movements between signs in a vocabulary and nonsign patterns (which include out-of-vocabulary signs, epentheses, and other movements that do not correspond to signs). In this paper, a novel method for designing threshold models in a conditional random field (CRF) model is proposed which performs an adaptive threshold for distinguishing between signs in a vocabulary and nonsign patterns. A short-sign detector, a hand appearance-based sign verification method, and a subsign reasoning method are included to further improve sign language spotting accuracy. Experiments demonstrate that our system can spot signs from continuous data with an 87.0 percent spotting rate and can recognize signs from isolated data with a 93.5 percent recognition rate versus 73.5 percent and 85.4 percent, respectively, for CRFs without a threshold model, short-sign detection, subsign reasoning, and hand appearance-based sign verification. Our system can also achieve a 15.0 percent sign error rate (SER) from continuous data and a 6.4 percent SER from isolated data versus 76.2 percent and 14.5 percent, respectively, for conventional CRFs.

Index Terms—Sign language recognition, sign language spotting, conditional random field, threshold model.

1 Introduction

Sign language recognition systems should be easy and natural to use. These systems should enable the user to gesture naturally in native sign language, without encumbrances of special clothing or tracking devices. In groundbreaking work in sign language recognition, many systems required the user to don special devices [14], [35]. Hand shape and motion are extracted easily and accurately using these devices. However, devices are expensive and, crucially, they reduce the naturalness of sign language communication.

One of the key tasks in sign language spotting is the task of detecting the start and end points of signs from continuous data and recognizing detected signs in a predefined vocabulary (see Fig. 1). In this paper, signs are movements in a predefined vocabulary and nonsign patterns are all other movements that are not included in the vocabulary (signs outside the set of vocabulary, and also other movements, like transitional movements and epentheses, that do not correspond to signs).

In this paper, we focus on machine vision methods for sign language spotting, i.e., detecting and recognizing signs in a known vocabulary, in videos of sentences and stories produced by native signers. The difficulty of sign language spotting is that instances of signs vary in both motion and appearance.

For dealing with motion, previous vision-based methods have demonstrated some successes using hidden Markov models (HMMs) [14], [32], [35] and conditional random fields (CRFs) [42], and so on [12], [13]. For spotting signs from continuous data, one HMM is constructed per label (sign) as shown in Fig. 2a. In contrast, one CRF model is constructed for all labels (signs) as shown in Fig. 2b. These previous approaches frequently employed a fixed threshold method for spotting or recognizing signs [14]. However, it is difficult to select a fixed threshold that is effective for all signs. As shown in Fig. 3a, if the threshold T_a is selected, then the sign "AND" is not spotted. In contrast, if the threshold T_b is selected, then the sign "AND" is correctly spotted. To solve this problem, work in speech recognition used a filler model or a garbage model, such as models based on HMMs [22], [30], [37]. However, it is difficult to obtain a representative set of nonsign patterns for training the filler model because a wide range of such patterns is possible. Lee and Kim proposed an HMM-based threshold model to overcome the weakness of the filler model [22]. Their threshold model is constructed by completely connecting states obtained from all gesture pattern HMMs. Although they reduced the number of states in the threshold model from 44 to 24 based on the relative entropy, the remaining number of states is still large and pattern discrimination is computationally expensive.

In this paper, we propose the augmentation of the CRF model by adding one additional label to overcome the weakness of the fixed threshold method (see Fig. 3b). A CRF is initially constructed without a label for nonsign patterns. Then, a CRF threshold model (T-CRF) is constructed by adding a label for nonsign patterns using the weights of

- *H.-D. Yang and S.-W. Lee are with the Department of Computer Science and Engineering, Asan Science Building, Room 238, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-713, Korea. E-mail: {hdyang, swlee}@image.korea.ac.kr.*
- *S. Sclaroff is with the Department of Computer Science, College of Arts and Sciences, Boston University, 111 Cummington St., Room 140E, Boston, MA 02215. E-mail: sclaroff@cs.bu.edu.*

Manuscript received 13 Feb. 2008; revised 10 June 2008; accepted 11 June 2008; published online 19 June 2008.
Recommended for acceptance by K. Murphy.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-02-0099.
Digital Object Identifier no. 10.1109/TPAMI.2008.172.

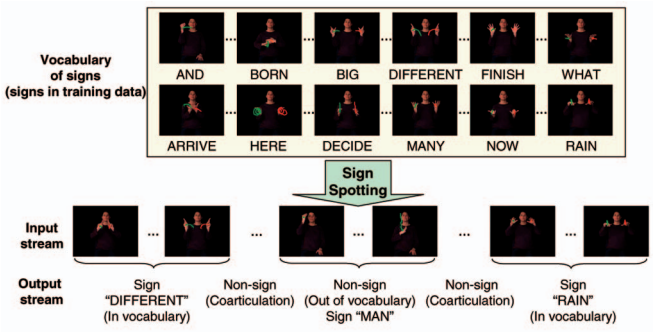

Fig. 1. An example of sign language spotting: Sign language spotting is the task of detecting the start and end points of signs in an input stream and recognizing detected signs in a predefined vocabulary.

state and transition feature functions of the original CRF. Therefore, the proposed CRF threshold model does not need nonsign patterns for training the model.

The previously mentioned models are useful for representing information about the overall hand trajectory of a sign. However, there are ambiguities among signs that exhibit similar overall hand movement and differ only in hand shape. The motion trajectories of some signs are very similar and this can lead to ambiguities in sign language spotting [17]. The hand shape can provide an important cue for verification of detected signs [1], [17]. Hand shapes at the start and end of a sign are called local features of the hand [33]. In our system, an appearance-based sign verification method is used to deal with hand shape [1].

To further improve spotting accuracy, a short-sign detector and a subsign reasoning method [2] are incorporated into our framework. The short-sign detector handles those signs that tend to have slower than the average

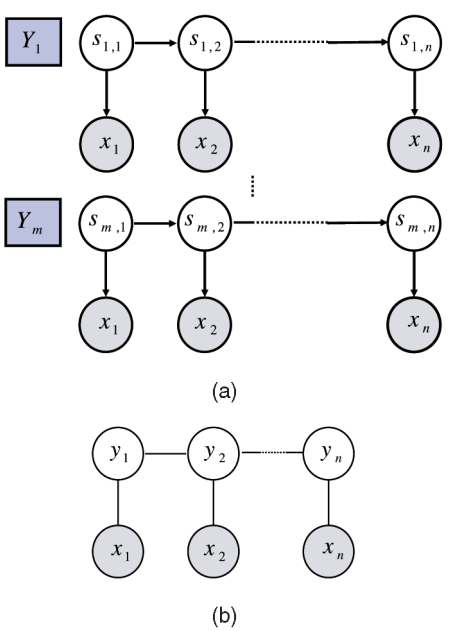

Fig. 2. Graphical representations of HMMs and CRFs. Y_i is a label, m is the number of labels, $s_{i,j}$ is a state of HMM for label Y_i, x_j is an observation of input sequence $\mathbf{x}$ at position j, y_j is a label of input sequence $\mathbf{x}$ at position j, $y_j \in \{Y_1, \ldots, Y_m\}$, and n is the length of observation sequence $\mathbf{x}$ [28]. (a) Structure of HMMs. (b) Structure of CRFs.

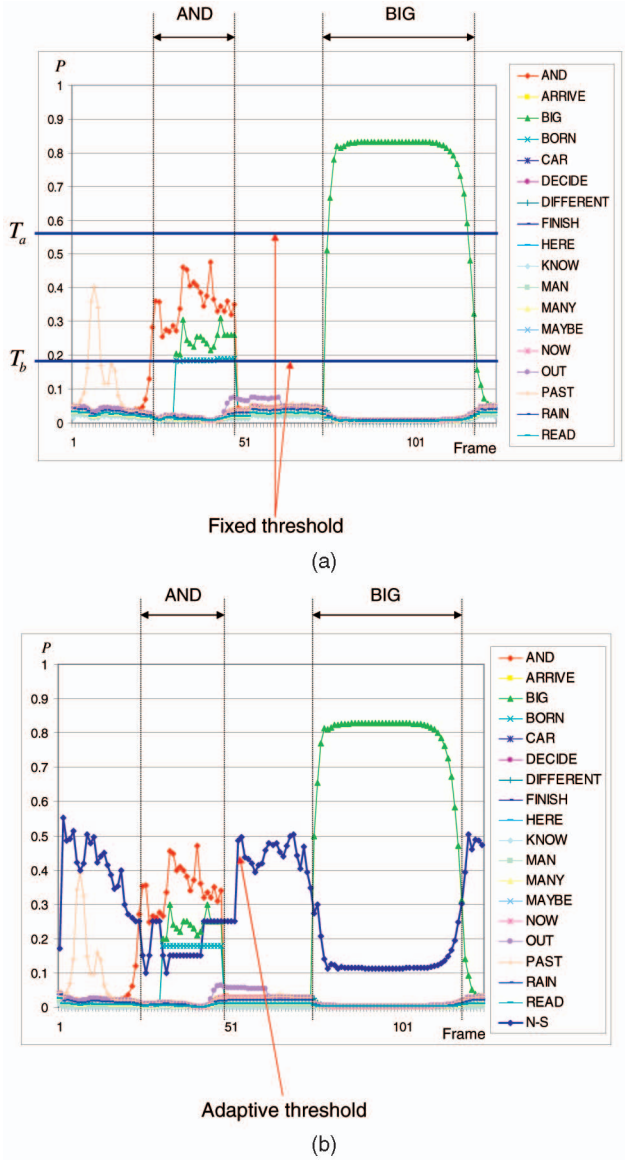

Fig. 3. The basic idea of the proposed threshold model for CRFs. N-S means a nonsign pattern (which includes out-of-vocabulary signs, epentheses, and other movements that do not correspond to signs). (a) Using a fixed threshold. (b) Using an adaptive threshold.

performance time. In our experiments, if the performance time for a specific sign is slower than the average performance time, the sign is classified as a short sign. It is more difficult to spot short signs than long signs because short signs have fewer samples than long signs. In our formulation, the weights of CRF self-transition feature functions of short signs are increased in order to enable accurate spotting of short signs.

Finally, following [2], a subsign reasoning method is employed to avoid premature detection of a sign that shares movements with other signs. For instance, a simple stroke sign may appear alone or as a part of a sequence of motions that form a single, more complex sign. In [2], a subgesture reasoning method was proposed for use in dynamic time warping (DTW). In our work, we adapt this subgesture reasoning method to a two-layer CRF architecture.

Fig. 4 shows a block diagram of the proposed sign language spotting method. A hand and face detector

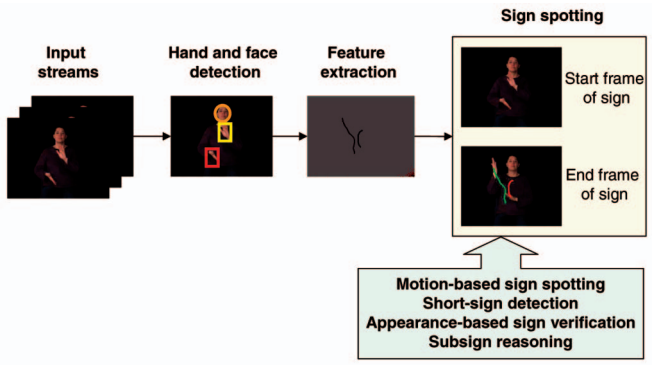

Fig. 4. Overview of the proposed sign language spotting method.

constitutes the preprocessing stage. A set of feature vectors is extracted according to the face position. Then, the sequence of feature vectors is analyzed in a sign language spotting algorithm.

The rest of this paper is organized as follows: Section 2 reviews related work, which is separated into two categories, pattern spotting and sign language recognition. Section 3 briefly reviews the basic CRF method. Section 4 gives a method for designing a threshold model for CRF. Section 5 presents a sign language spotting algorithm. Section 6 presents experimental results and discusses their implications. Section 7 concludes this paper.

2 Related Work

Pattern spotting is the task of extracting meaningful segments from input signals and recognizing them in a predefined set of data [1], [2], [22]. Sign language spotting is an instance of pattern spotting [1]. There are many pattern spotting algorithms and applications. The representative pattern spotting algorithms and sign language systems are briefly reviewed in this section. For a review of pattern spotting methods for sequential data, refer to the review paper by Dietterich [11]. For a comprehensive review of sign language analysis, refer to the survey paper by Ong and Ranganath [26].

2.1 Pattern Spotting

Pattern spotting is a very important area of speech recognition, DNA sequences analysis, and computer vision research. A pattern spotting algorithm is required in order to find predefined patterns in the input data.

Template matching-based approaches identify matches between the input data and a set of predefined templates. The DTW is a widely used algorithm for computing the similarity between the input data and a template. The DTW has been successfully used for speech recognition and hand gesture spotting [1], [3], [18].

The HMM is a widely used model for spotting patterns with spatiotemporal variability. This has been successfully applied in several sign language recognition systems, e.g., [14], [32], [35]. HMMs comprise a set of states S, a set of observations O, and two conditional probability distributions; one is the state transition probability and the other is the observation probability. The state transition probability is the probability of passing from state s' to state s, $P(s|s')$ for s, $s' \in S$, the observation probability is $P(o|s)$ for $o \in O$, and the set of initial state probabilities is $P_0(s)$ [21], [23], [29].

McCallum et al. proposed maximum entropy Markov models (MEMMs), which use a directed graphical model [23]. MEMMs combine transition and observation probabilities of HMMs into a single probability $P(s|s', o)$. The current observation dependency differs between HMMs and MEMMs. The current observation in HMMs only depends on the current state, but the current observation in MEMMs depends on the current and previous states [23].

MEMMs suffer from the label bias problem. This is due to the fact that MEMMs are locally normalized. Lafferty et al. proposed CRFs, which use an undirected graphical model to overcome the weakness of MEMMs [21]. The difference between CRFs and MEMMs is that MEMMs use per-state exponential models for conditional probabilities of the next state given the current state, whereas CRFs use a single model for the joint probability of the sequence of labels given the observation sequence. Thus, there are trade-offs in the weights of each feature function, for each state [21].

Hidden CRFs (HCRFs) are an extension of CRFs that include hidden variables [15], [28], [34]. HCRFs automatically model the local interconnection between parts. HCRFs enable the sharing of information between labels with hidden variables. Therefore, HCRFs cannot model dynamics between labels. HCRFs can estimate a class given a segmented sequence [28].

Morcency et al. proposed latent-dynamic hidden CRFs (LDCRFs) [24]. LDCRFs are a framework for detecting and recognizing sequential data, which can model the substructure of a label and learn dynamics between labels [24]. Therefore, LDCRFs can detect and recognize labels from unsegmented data.

2.2 Sign Language Spotting and Recognition

Automatic sign language recognition systems can be classified into two approaches according to data acquisition method, data acquired via direct-measurement devices (device-based) or via cameras (vision-based) [26]. Sign language recognition can also be separated into two classes according to test data, a continuous gesture data set or an isolated gesture data set. Many previous approaches focused on isolated sign language recognition because isolated sign language recognition is easier than continuous sign language recognition [1].

2.2.1 Device-Based Approaches

The hand shape and motion are extracted easily using direct-measurement devices such as data gloves and position trackers. Thus, device-based approaches are generally more accurate and can recognize a wider vocabulary than vision-based approaches.

Braffort proposed a French Sign Language recognition system using HMMs [7]. A data glove was used to obtain hand appearance and position. Features of hands are extracted from their appearance and position. Signs are separated into conventional signs, nonconventional signs, and variable signs. Two classifiers were used to recognize signs; one was used to recognize conventional signs and the other was used to recognize nonconventional signs and variable signs. The system achieved a recognition rate of 96 percent with a vocabulary of seven signs. The experiments were performed with a continuous gesture data set.

Gao et al. proposed a Chinese Sign Language recognition system, which was based on a dynamic programming

method [14]. The system used two data gloves and three position trackers to extract the hand appearance and position. The DTW method was used to match an input data and templates of movement epentheses. The system could recognize 5,113 signs and achieved a recognition rate of 90.8 percent with 1,500 sentences. The experiments were performed with a continuous data set. It was assumed that movement epentheses between two signs are always similar in different sentences. However, movement epentheses between two signs vary in real-world applications.

Vogler and Metaxas described an American Sign Language (ASL) recognition system based on HMMs [35]. Three video cameras were used. An electromagnetic tracking system was used to extract 3D parameters of the signer's arm and hand. Two experiments were performed, both with 99 test sentences and a vocabulary of 22 signs. The system achieved a recognition rate of 94.5 percent for single signs. It achieved a recognition rate of 84.5 percent for complete sentences. The experiments were performed with a continuous data set.

2.2.2 Vision-Based Approaches

Vision-based sign language recognition approaches need hand detection and tracking algorithms to extract hand locations. Color, motion, or edge information is generally used to detect hands from input data. Vision-based approaches have limitations according to imaging conditions such as background, illumination, clothing, and so on [1].

Starner et al. presented an ASL recognition system based on HMMs [32]. The experiments involved two systems using 40 signs. The first system observed the signer using a camera mounted on a desk and achieved 92 percent accuracy. The second system observed the signer using a camera mounted on a cap worn by the signer and achieved 98 percent accuracy. Their experiments were performed with a continuous data set. Only hand motion features were used; hand shapes to disambiguate signs with similar hand motions were not used.

Bauer and Kraiss proposed a German Sign Language recognition system based on HMMs in which the signer wore simple colored gloves to obtain data [5]. Subunit HMMs were used to model signs. Two experiments were performed. In the first experiment, a 92.5 percent recognition rate was achieved for 100 signs. In the second experiment, subunit HMMs used in the first experiment also performed the sign language spotting task, without retraining for 50 new signs. A recognition rate of 81.0 percent was achieved with an isolated data set.

Holden et al. proposed an Australian Sign Language recognition system based on HMMs [16]. The system extracted the angle between two hands with respect to the head, their directions of movement, roundedness, and size ratio. A 97 percent recognition rate at the sentence level and 99 percent at the word level were achieved. The experiments were performed with a continuous data set.

Bowden et al. developed a British Sign Language recognition system based on Markov chains combined with independent component analysis [6]. The system extracted a feature set describing the location, motion, and shape of hands. A recognition rate of 97 percent was achieved with 43 signs.

Yang et al. proposed an ASL recognition system based on a time-delay neural network [41]. The system used motion information to extract hand positions. The recognition rate was 96.2 percent with 40 signs. The experiments were performed with an isolated data set.

Yang and Sarkar proposed an ASL spotting method based on CRFs [42]. The system used motion information as features and the Kanade-Lucas-Tomasi method to track the motion of salient corner points. The system extracted key frames from sentences in the training data. Then, the system labeled each key frame with a coarticulation pattern or sign pattern. The spotting rate was 85 percent with 39 signs, articulated in 25 different sentences.

Yang et al. proposed an ASL recognition method based on an enhanced Level Building algorithm [43]. An enhanced classical Level Building algorithm was used, based on a dynamic programming approach to spot signs without explicit coarticulation models. The recognition rate was 83 percent with 39 signs, articulated in 25 different sentences. This research only considered hand motion. It did not consider hand shape for recognizing signs.

Nayak et al. proposed an ASL recognition method based on a continuous state space model [25]. They used an unsupervised approach to extract and learn models for continuous basic units of signs, which are called signemes, from continuous sentences. They automatically extracted a signeme model given a set of sentences with one common sign.

As mentioned in Section 1, one HMM is constructed per label. Therefore, HMMs assume that all the observations are independent. In CRFs, one model is constructed for all labels. Therefore, CRFs assume that every observation is related for all labels. Thus, there is a trade-off in the number of occurrences of a feature value for each label.

Few researchers have addressed the problem of nonsign patterns (which include out-of-vocabulary signs, epentheses, and other movements that do not correspond to signs) for sign language spotting [14], [32], [42], [43]. This is because it is difficult to model nonsign patterns. As mentioned previously, Gao et al. assumed that movements between two signs are always similar in different sentences [14]. Yang and Sarkar extracted key frames for training coarticulatory movements from sentences consisting of in-vocabulary signs and nonsign patterns [42].

As mentioned in Section 1, hand shape and motion are important features for recognizing sign language. However, many previous approaches only used hand motion for recognizing sign language [14], [43].

In this paper, we propose a CRF threshold model. The proposed threshold model can distinguish in-vocabulary signs and nonsign patterns. Nonsign patterns are thereby modeled by the threshold model without a training set for nonsign patterns. Also, hand and motion information is used to spot signs.

3 Conditional Random Fields

CRFs are a framework based on conditional probability approaches for segmenting and labeling sequential data [21], [36]. CRFs use a single exponential distribution to model all labels of given observations. Therefore, there is a trade-off in the weights of each feature function, for each state [21]. In our application, each state corresponds to a sign in a vocabulary.

3.1 CRF Framework

In CRFs, the probability of label sequence $\mathbf{y}$, given observation sequence $\mathbf{x}$, is found using a normalized product of potential functions. Each product of potential functions is represented by [21], [36]

$$\exp\left(\sum_v \lambda_v t_v(y_{i-1}, y_i, \mathbf{x}, i) + \sum_m \mu_m s_m(y_i, \mathbf{x}, i)\right), \quad (1)$$

where $t_v(y_{i-1}, y_i, \mathbf{x}, i)$ is a transition feature function of observation sequence $\mathbf{x}$ at positions i and $i-1$, $s_m(y_i, \mathbf{x}, i)$ is a state feature function of observation sequence $\mathbf{x}$ at position i, y_{i-1} and y_i are labels of observation sequence $\mathbf{x}$ at positions i and $i-1$, and λ_v and μ_m are weights of transition and state feature functions, respectively.

A state feature function indicates whether a feature value is observed at a particular label or not. The state feature function is defined by

$$s_m(y_i, \mathbf{x}, i) = \begin{cases} b(\mathbf{x}, i), & \text{if } y_i = Y_a, \\ 0, & \text{otherwise}, \end{cases} \quad (2)$$

where Y_a is a label of a CRF, y_i is a label of observation sequence $\mathbf{x}$ at position i, and $b(\mathbf{x}, i)$ is defined by

$$b(\mathbf{x}, i) = \begin{cases} 1, & \text{if the feature value of observation} \\ & \text{sequence } \mathbf{x} \text{ at position } i \text{ is } r, \\ 0, & \text{otherwise}, \end{cases} \quad (3)$$

where r is a feature value.

A transition feature function indicates whether a feature value is observed between two states or not. The transition feature function is defined by

$$t_v(y_{i-1}, y_i, \mathbf{x}, i) = \begin{cases} b(\mathbf{x}, i), & \text{if } y_{i-1} = Y_a \text{ and } y_i = Y_b, \\ 0, & \text{otherwise}, \end{cases} \quad (4)$$

where Y_a and Y_b are labels of a CRF and y_{i-1} and y_i are labels of observation sequence $\mathbf{x}$ at positions i and $i-1$, respectively.

From (1), the probability of label sequence $\mathbf{y}$, given observation sequence $\mathbf{x}$, is calculated by

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\theta(\mathbf{x})} \exp\left(\sum_{i=1}^{n} F_\theta(y_{i-1}, y_i, \mathbf{x}, i)\right), \quad (5)$$

where parameter $\theta = (\lambda_1, \lambda_2, \ldots, \lambda_{N_T}; \mu_1, \mu_2, \ldots, \mu_{N_S})$, N_T is the number of transition feature functions, N_S is the number of state feature functions, n is the length of observation sequence $\mathbf{x}$,

$$F_\theta(y_{i-1}, y_i, \mathbf{x}, i) = \sum_v \lambda_v t_v(y_{i-1}, y_i, \mathbf{x}, i) + \sum_m \mu_m s_m(y_i, \mathbf{x}, i),$$

and $Z_\theta(\mathbf{x})$ is the normalization factor

$$Z_\theta(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left(\sum_{i=1}^{n} F_\theta(y_{i-1}, y_i, \mathbf{x}, i)\right). \quad (6)$$

3.2 Learning CRF Parameter

CRF parameter learning is based on the principle of maximum entropy. Maximum likelihood training selects parameters that maximize the log-likelihood of the training data. The log-likelihood for a CRF is

$$\begin{aligned} \mathcal{L}(\theta) &= \sum_{t=1}^{T} \log p_\theta(\mathbf{y}^t|\mathbf{x}^t) \\ &= \sum_{t=1}^{T} \left(\sum_{i=1}^{n} F_\theta(y_{i-1}, y_i, \mathbf{x}^t, i) - \log Z_\theta(\mathbf{x}^t)\right), \end{aligned} \quad (7)$$

where T is the number of training sequences, $\mathbf{x}^t$ is an observation sequence of the training set, and $\mathbf{y}^t$ is the corresponding label sequence for observation sequence $\mathbf{x}^t$.

There is no closed form solution to (7). Instead, a solution is found via iterative techniques [21], [36]. Likelihood maximization is performed using a gradient-based method, where

$$\begin{aligned} \frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \sum_{t=1}^{T} \Bigg(& \sum_{i=1}^{n} \frac{\partial F_\theta(y_{i-1}^t, y_i^t, \mathbf{x}^t, i)}{\partial \theta} \\ & - \sum_{\mathbf{x}} p_\theta(\mathbf{y}|\mathbf{x}^t) \sum_{i=1}^{n} \frac{\partial F_\theta(y_{i-1}, y_i, \mathbf{x}^t, i)}{\partial \theta} \Bigg). \end{aligned} \quad (8)$$

3.3 Matching CRF

The probability $p_\theta(\mathbf{y}|\mathbf{x})$ of label sequence $\mathbf{y}$, given observation sequence $\mathbf{x}$, is computed using matrices [36]. Two dummy states, start $\mathbf{y}_0$ and stop $\mathbf{y}_{n+1}$, are added [21], [31], [36]. A set of $n+1$ matrices $\{M_i(\mathbf{x}) | i = 1, \ldots, n+1\}$ is computed. Each $M_i(\mathbf{x})$ is an $|S \times S|$ matrix with elements of the form [31], [36]

$$M_i(y', y|\mathbf{x}) = \exp(F_\theta(y', y, \mathbf{x}, i)), \quad (9)$$

where $S = \{Y_1, \ldots, Y_l\}$ is a set of labels of the training data, Y_i is a label of the training data, l is the number of labels, and y and y' are labels of S at time i.

The probability $p_\theta(\mathbf{y}|\mathbf{x})$ of label sequence $\mathbf{y}$, given observation sequence $\mathbf{x}$, is computed via the product of $n+1$ matrices

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{\prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|\mathbf{x})}{Z_\theta(\mathbf{x})}. \quad (10)$$

The normalization factor $Z(\mathbf{x})$ for observation sequence $\mathbf{x}$ is computed from the set of $M_i(\mathbf{x})$ matrices [31], [36]. The normalization factor $Z(\mathbf{x})$ is computed via the product of all matrices:

$$Z_\theta(\mathbf{x}) = \left(\prod_{i=1}^{n+1} M_i(\mathbf{x})\right)_{\text{start, stop}}. \quad (11)$$

Algorithm 1 gives the method for matching an input with a CRF model.

Algorithm 1 CRF matching algorithm

input An observation sequence $\mathbf{x}$ with length O_l
output The probability $p_\theta(\mathbf{y}|\mathbf{x})$ of the label sequence $\mathbf{y}$ given the observation sequence $\mathbf{x}$

1: **for** $i = 1$ to O_l **do**
2: **for** $j = 1$ to l **do**
3: **for** $k = 1$ to l **do**
4: $M_i(y_j, y_k) = \exp F_\theta(y_j, y_k, \mathbf{x}, i)$ {Calculate potential functions}
5: **end for**
6: **end for**

7: $Z = Z \times M_i$ {Calculate normalization factor $Z(\mathbf{x})$ for the observation sequence $\mathbf{x}$}
8: **end for**
9: **for** $i = 1$ to O_l **do**
10: $Unnormalized\_p^* = M_i(y_{i-1}, y_i|\mathbf{x})$
11: **end for**
12: $p_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \times Unnormalized\_p^*$ {Calculate the probability of the label sequence $\mathbf{y}$ given the observation sequence $\mathbf{x}$}

4 Threshold Model with Conditional Random Fields

Many existing sign detection methods employ a fixed threshold that best discriminates in-vocabulary signs and nonsign patterns (which include out-of-vocabulary signs, epentheses, and other movements that do not correspond to signs) [32], [43]. However, it is difficult to select a fixed threshold that is effective for all labels. To address this problem, we propose the following approach.

In our formulation, a CRF is initially constructed using the method described in Section 3 without a label for nonsign patterns. Then, a threshold model with CRF (T-CRF) is constructed by adding the label for nonsign patterns G in the original CRF using the weights of state and transition feature functions of the original CRF. Therefore, labels of the T-CRF are $S_T = \{Y_1, \ldots, Y_l, G\}$, where l is the number of labels of the CRF and G is the label for nonsign patterns, as described below.

4.1 Weight of State Feature Function

As mentioned in Section 3, a CRF uses a single model for the joint probability of the sequence. Therefore, there is a trade-off between the weights of each feature function, for each state [21]. In a CRF, the weights of a state feature function are distributed over several labels when its variance is small, whereas the weights of a state feature function are concentrated at specific labels when its variance is large. By assigning the weight of a state feature function of the label for nonsign patterns G according to the variance of weights of the feature functions, the threshold model with CRF can correctly spot in-vocabulary signs and nonsign patterns.

Dugad et al. proposed a threshold model based on the standard deviation and mean of samples [19]. The weight of a state feature function of the label for nonsign patterns G is calculated by applying Dugad et al.'s method:

$$\mu_m(G) = \bar{\mu}_m + T_d\sqrt{\sigma_{\mu_m}}, \tag{12}$$

where $\bar{\mu}_m = \frac{\sum_{k=1}^{l} \mu_m(Y_k)}{l}$ and σ_{μ_m} is the variance of weights of the mth feature function.

We compute the best T_d for maximizing the overall recognition rate with the training set (determined by experiment in our system, $T_d = 1.8$).

4.2 Weight of Transition Feature Function

A CRF is dominated by the label with the maximum weight in feature functions. In sign language sentences, frequencies of nonsign patterns are larger than those of in-vocabulary signs. Therefore, the weights for feature functions of the label for nonsign patterns G should be higher than those of the other labels (signs).

The weight of the self-transition feature function of the label for nonsign patterns G is calculated by

$$\lambda_v(G, G) = \underset{k=1,\ldots,l}{\operatorname{argmax}} \lambda_v(Y_k, Y_k) + \frac{S_{ff}}{\bar{N}_{ff}}, \tag{13}$$

where $\frac{S_{ff}}{\bar{N}_{ff}}$ is the weight of a self-transition feature function of the label for nonsign patterns G, λ_v is the transition feature function weight, $\bar{N}_{ff}$ is the average number of features in which the weight is larger than 0, and $C = \sum_{k=1}^{l} \sum_{m=1}^{N_s} \mu_m(Y_k)$ is the sum of all state feature function weights.

Lee and Kim [22] and Yang et al. [39] proposed simple methods to design a threshold model based on HMMs. The transition probability between states of the threshold model is assigned as $1/n$, where n is the number of states of the threshold model. In our CRF model, a similar method is applied for calculating weights of transition feature functions between the label for nonsign patterns and the other labels.

In an observation sequence of signs, if the label of the current frame of the observation sequence is a specific sign, then the label of the next frame is a sign or a nonsign pattern. Only one frame is used as a transition frame from a specific sign to another sign or a nonsign pattern. Therefore, weights of transition feature functions from the other labels to the label for nonsign patterns G are assigned by

$$\forall_{k\in\{1,\ldots,l\}}\ \lambda_v(Y_k, G) = \frac{\lambda_v(Y_k, Y_k)}{l}. \tag{14}$$

Nonsign patterns can be observed at any time and they can undergo a transition to a specific sign or a nonsign pattern. Therefore, weights of transition feature functions from the label for nonsign patterns G to the other labels are assigned by

$$\forall_{k\in\{1,\ldots,l\}}\ \lambda_v(G, Y_k) = \frac{\lambda_v(G, G)}{l}. \tag{15}$$

4.3 Spotting with Threshold Model

The Viterbi algorithm is used to spot signs in a given observation sequence. Given an observation sequence $\mathbf{x} = x_1, x_2, \ldots, x_n$ and likelihood $p_\theta(\mathbf{y}|\mathbf{x})$, the best state sequence $\mathbf{y} = y_1, y_2, \ldots, y_n$ is found by

$$\delta_t(i) = \max_{y_1, y_2, \ldots, y_{t-1}} p_\theta[y_1, y_2, \ldots, y_t = i|x_1, x_2, \ldots, x_t], \tag{16}$$

where $y_i \in \{Y_1, \ldots, Y_l, G\}$.

Equation (16) accounts for the first t observations of the input data. The end state is y_i. From (5) and (16), we can discover the general induction

$$\delta_{t+1}(i) = \max_i[\delta_t(i) F_\theta(y_{i-1}, y_i, \mathbf{x}, i)]. \tag{17}$$

The start and end points of the sign are obtained by keeping the best state calculated by (17) to the current state, e.g.,

$$\psi_t(i) = \underset{1\le j\le n}{\operatorname{argmax}}[\delta_t(j)]. \tag{18}$$

If we examine the time evolution of the likelihood at each level, the level for nonsign patterns usually has the best score. As a forward pass gets close to the end of a sign, the probability of the sign falls below those of the other levels.

As shown in Fig. 10, the probability of the sign "AND" is initially lower than that of the N-S (nonsign pattern). After 25 frames, however, the probability of the sign "AND" becomes greater. After 46 frames, the probability of the sign "AND" nearly becomes zero and there is a nonsign pattern. The candidate end points of the sign "AND" are between frames 47 and 175. As mentioned in [22], end-point detection is the process of selecting the best point to distinguish each sign. Detection criteria are defined as follows:

- If the subsequent sign is a nonsign pattern, then the start point of the current nonsign pattern is determined as the end point of the previous sign.
- If the subsequent sign is a sign, then there are two cases:
  - If the previous sign is a part of the current sign (see Section 5.4), then the previous sign is considered as a part of the current sign.
  - If the previous sign and current sign are different, then the start frame of the current sign is determined as the end point of the previous sign. A delayed response is observed in this case.

Algorithm 2 gives the method for detecting the start and end points of signs.

Algorithm 2 End-point detection algorithm
input The probability of the threshold model with CRF T_CRF, an observation sequence $\mathbf{y}$ with length O_l
output The start and end points of signs
1: **for** $i = 1$ to O_l **do**
2: $Current\_Sign = \max T\_CRF(i)$ {Select the sign with maximum probability}
3: **if** $Is\_Nonsign(Current\_Sign)$ {Check whether the current sign is a nonsign pattern or not} **then**
4: **if** $Stack.size() = 1$ **then**
5: $Spotted\_Sign = Stack.pop()$
6: $Start\_Point = Backtracking(Spotted\_Sign)$
7: $End\_Point = i$ {Spot a sign}
8: **else**
9: **if** $Is\_SubSign(Stack)$ **then**
10: **while** $!Is\_Empty(Stack)$ **do**
11: $Spotted\_Sign = Stack.pop()$ {Ignore subsigns}
12: **end while**
13: $Start\_Point = Backtracking(Spotted\_Sign)$
14: $End\_Point = i$ {Spot a sign considering subsign reasoning}
15: **else**
16: **while** $!Is\_Empty(Stack)$ **do**
17: $Spotted\_Sign = Stack.pop()$
18: $Start\_Point = Backtracking(Spotted\_Sign)$
19: $End\_Point = Spotted\_Sign.End$ {Spot signs}
20: **end while**
21: **end if**
22: **end if**
23: **else if** $Current\_Sign! = Previous\_Sign$ **then**
24: $Sign.End = i$
25: $Sign.Label = Current\_Sign$
26: $Stack.push(Sign)$ {Push a sign on the stack}
27: **end if**
$Previous\_Sign = Current\_Sign$
28: **end for**

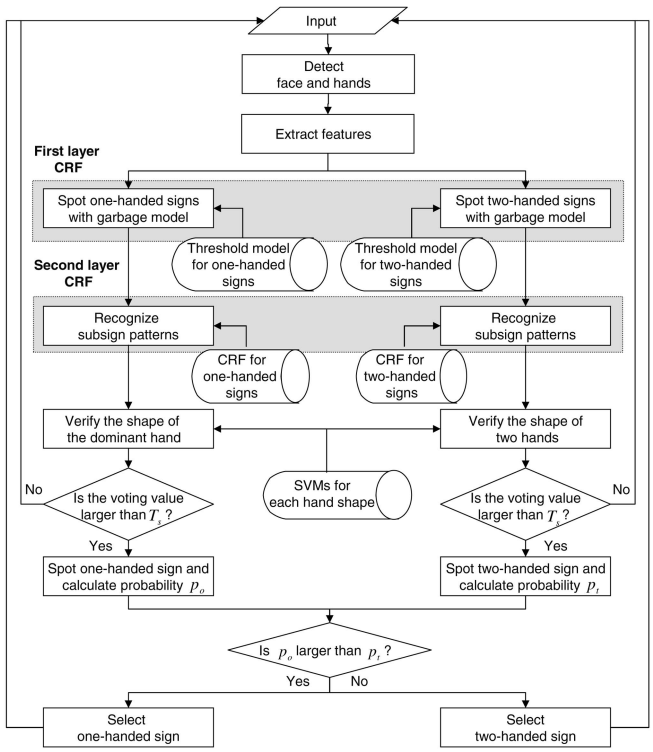

Fig. 5. Flowchart of the proposed sign language spotting method.

The most likely state sequence for a sign in the input sequence is identified by backtracking of the Viterbi path by following the chain of back pointers $\psi_t()$. The threshold model with CRF can distinguish in-vocabulary signs and nonsign patterns by selecting the label with the maximum probability.

5 Sign Language Spotting System

In order to spot signs from continuous data, a two-layer CRF architecture is applied with a subsign reasoning method, a short-sign detector, and an appearance-based shape verification method. Fig. 5 shows the flowchart of the proposed ASL spotting system.

In the first layer, in-vocabulary signs and nonsign patterns are discriminated by a threshold model with CRF. Spotted signs in the first layer are temporarily saved. Subsequent to detecting a sign sequence, the second layer CRF is applied to find subsign patterns. Subsign patterns within signs are modeled with a CRF. Table 3 shows sample data for training the second layer CRF. The input sequence of the second layer CRF includes more than one sign. The results of the second layer CRF are used to find the candidate label with the maximum probability. Finally, the appearance-based sign verification method is performed for the selected candidate sign.

5.1 Hand Detection and Tracking

Color and motion information are used to detect hand regions. A face detector [38] is used to detect the signer's face. The skin region is detected via a simple color threshold

TABLE 1
Two Motion-Based and Four Location-Based Features

Features	Meanings
P_{LH}	Position of the left hand
P_{RH}	Position of the right hand
S_{TH}	Vertical symmetry of two hands
O_{TH}	Occlusion of two hands
C_{LH}	Directional codeword between the current and the previous positions of the left hand
C_{RH}	Directional codeword between the current and the previous positions of the right hand

method. A pixel is classified as skin if its H and S color values are within the standard deviation of the mean skin color value in HSV space. A frame differencing method is used to detect hand motion. The detected skin color regions and motion regions are then multiplied together.

An appearance-based hand tracking algorithm is used to track the hand region [40]. If the hand detector successfully extracts the hand region, then the hand appearance is stored as a template. The template is used if the detected hand region is much larger than the hand region detected in the previous frame or the hand detector fails to detect the hand region. Given a template R containing the hand region detected in the previous frame, the best match is determined in the current frame by performing correlation matching. In order to evaluate displacement d_x, d_y for the new hand region, the sum of differences in absolute intensity between pixels in region R and corresponding pixels $x + d_x, y + d_y$ in the current frame is calculated by

$$D(d_x, d_y) = \sum_R \left| I_{t-1}(x, y) - I_t(x + d_x, y + d_y) \right|, \tag{19}$$

where I_t is an intensity image at time t and $I_t(x, y)$ is the intensity value at position x, y in the image I_t.

All offsets d_x, d_y are calculated. The position of the best match $\widehat{d}_x, \widehat{d}_y$ is given by

$$\widehat{d}_x, \widehat{d}_y = \min_{d_x, d_y} D(d_x, d_y). \tag{20}$$

5.2 Motion-Based and Location-Based Sign Language Spotting

Given hand and face regions detected in a frame, six features are extracted as shown in Table 1. A CRF is constructed using six features.

The feature P_{LH} is extracted by clustering a feature vector $\{\theta_{FLH}, d_{FLH}\}$ into an index, where θ_{FLH} is the angle between the center of the face and the center of the left hand, and d_{FLH} is the distance between the center of the face and the center of the left hand (see Fig. 6a). Clustering of the set of feature vectors $\{\theta_{FLH}, d_{FLH}\}$ is found via an EM-based Gaussian mixture model (GMM). We determined via experiments that the optimum number of clusters for each hand is in the range of 30-36.

The hand symmetry is calculated by

$$S_{TH} = \begin{cases} 1, & |d_{HL} - d_{HR}| < T_h \text{ and } d_V < T_v, \\ 0, & \text{otherwise}, \end{cases} \tag{21}$$

where d_{HL} and d_{HR} are the horizontal distances between the center of the face and the centers of the left and right hands, respectively. d_V is the vertical distance between the centers

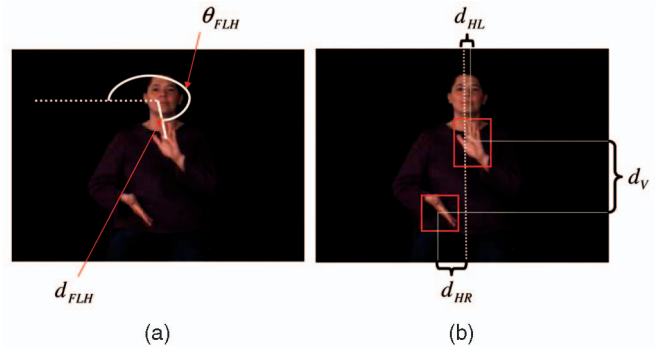

Fig. 6. (a) Representation of hand position and (b) symmetry of hands.

of the hands and T_h and T_v are distance thresholds for the horizontal and vertical differences between the hands, respectively (see Fig. 6b). We determined in our experiments that the most effective thresholds are approximately 10 and 12, using adult signers at an image resolution of 320×240 pixels.

The hand occlusion is determined by calculating the ratio of the overlapped region of two hands

$$O = \begin{cases} 1, & \min\left(\frac{R_o}{H_l}, \frac{R_o}{H_r}\right) > T_o, \\ 0, & \text{otherwise}, \end{cases} \tag{22}$$

where H_l is the left hand region, H_r is the right hand region, R_o is the overlapped region between two hands, and T_o is the threshold for the hand occlusion (determined by experiment, $T_o = 0.3$).

The directional codewords C_{LH}, C_{RH} are set to one of eight direction codewords or a dummy codeword. The dummy codeword represents the case where there is negligible movement between two positions [22].

Initially, a CRF is trained using six features shown in Table 1. Then, a threshold model with CRF is constructed using the weights of state and transition feature functions of the CRF by augmenting the CRF with one additional label that can play the role of an adaptive threshold as described in Section 4. Two threshold models with CRF are constructed to spot signs using hand movements, one for one-handed signs and the other for two-handed signs as shown in Fig. 5. The start and end points of all signs are obtained by backtracking of the Viterbi path subsequent to a forward pass, as described in Section 3.

5.3 Short-Sign Detector

It is more difficult to spot short signs than long signs because short signs have fewer samples than long signs. To overcome this problem, the weights of self-transition feature functions for short signs are increased:

$$\lambda_v(Y_k, Y_k) = \begin{cases} \lambda_v(Y_k, Y_k) + w_v(Y_k), & \text{if } N_{FRv}(Y_k) < (\bar{N}_{FR} - \sigma_{N_{FR}}) \\ \lambda_v(Y_k, Y_k), & \text{otherwise}, \end{cases} \tag{23}$$

where $\bar{N}_{FR}$ is the average performance time, $\sigma_{N_{FR}}$ is the variance of performance time, $N_{FRv}(Y_k)$ is the average performance time at label Y_k, and $w_v(Y_k)$ is the weight of label Y_k, i.e.,

TABLE 2
Short and Long Signs Detected in the Vocabulary

Short signs	$N_{FR}(Y_k)$	Long signs	$N_{FR}(Y_k)$
AND	16	BORN	33
ARRIVE	20	BOY	26
BICYCLE	23	BUTTER	28
BIG	13	DAY	26
BLACK	12	CAR	63
DECIDE	16	COLD	28
DIFFERENT	13	HERE	67
FARM	24	INTERPRET	26
INFORM	20	FUNNY	26
FINISH	14	LIBRARY	52
GOOD	22	MAGAZINE	40
HOT	20	KNOW	26
MANY	22	MAYBE	57
LIE	22	NAME	30
LIKE	24	NIGHT	28
MAN	24	RAIN	39
NOW	12	READ	25
OUT	15	SHOES	30
PAST	13	SORRY	42
SIT	22	TAKE-OFF	25
STRANGE	20	WHAT	42
WANT	22	WORK	30
TELL	14	YESTERDAY	26
TOGETHER	11	WOW	47

$N_{FR}(Y_k)$ is the average performance time of sign Y_k

$$w_v(Y_k) = \frac{(\bar{N}_{FR} - \sigma_{N_{FR}}) - N_{FR_v}(Y_k)}{\text{argmax}_{k=1,\ldots,l} N_{FR_v}(Y_k)}. \quad (24)$$

Table 2 shows short signs detected in the training data set shown in Table 5. The average performance time of all signs is 25 frames captured at a rate of 30 frames/s.

5.4 Subsign Reasoning

Some signs appear as parts of other signs. A two-layer CRF architecture is applied to deal with this problem. The first layer uses a threshold model with CRF. The second layer uses a CRF. In the first layer, the threshold model with CRF discriminates in-vocabulary signs and nonsign patterns. Then, in the second layer, subsign patterns are recognized in the spotted sign sequence by the first layer.

To find shared patterns among signs, only the first layer CRF is applied to training sequences. An error in the middle of a sign implies that the sign has been confused with another sign in the vocabulary or an improper temporal boundary has been detected. Errors such as the confusion between signs are due to the problem of shared patterns among signs [1]. Fig. 7 shows the problem of shared patterns among signs in the sign language spotting algorithm. As shown in Fig. 7, part of the sign "DECIDE" is confused with the sign "HERE" in the region of the ground truth.

The second layer CRF models such subsign patterns. An input observation of the second layer CRF $\mathbf{x}'$ is a sequence of distinct signs, which is discriminated by the first layer CRF. For example, in Fig. 7, the input observations of the second layer CRF are "DECIDE" and "HERE." Table 3 shows the extracted subsign patterns. The second layer CRF is trained using the data in Table 3. Subsequent to training the second layer CRF, signs are selected with a normalized probability of the input sequence.

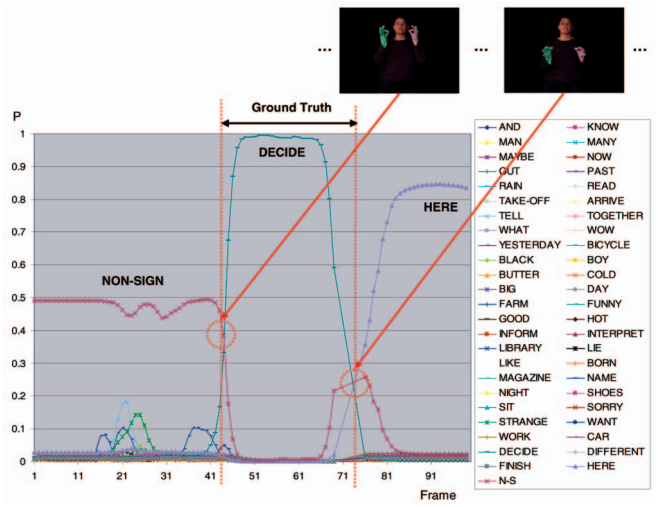

Fig. 7. Temporal evolution of probabilities of signs including subsign patterns; the vertical dashed line marks the ground truth of sign "DECIDE."

5.5 Appearance-Based Sign Verification

The main goal of the appearance-based sign verification method is to decide whether or not to accept a sign spotted via motion-based and location-based sign spotting methods. This helps to disambiguate signs that may have similar overall hand motions but different hand shapes.

One SVM per hand shape class is constructed in precomputation, for use in the online verification step. In our method, only the hand shape at the start of the sign is verified. For one-handed signs, only the dominant hand's shape is verified. For two-handed signs, both the dominant and nondominant hands' shapes are verified.

The appearance-based sign verification classifier is trained with a set of images of a particular hand shape (positive samples) and a set of arbitrary images extracted from the background and other hands (negative samples) [1]. Hand shapes were generated via POSER [27]. Table 4 shows examples of hand shapes. Histograms of gradient features are extracted for training SVM from training examples [10].

The hand appearance is verified over a period of several frames. Then, voting over t_a frames is used to decide whether to accept or reject the sign. If voting result v is larger than threshold T_s (determined by experiment, $T_s = 3$), then the candidate sign is determined to be a sign to be labeled. The start and end points of the sign are

TABLE 3
Examples of Subsign Patterns Extracted in the Training Data

Supersigns	Subsigns ($\mathbf{x}'$)
AND	{TELL, AND}, {PAST, AND}
ARRIVE	{ARRIVE, DIFFERENT}, {ARRIVE, TOGETHER}
BICYCLE	{BICYCLE, CAR}
BIG	{BIG, MANY}, {BIG, DECIDE}
BUTTER	{BUTTER, READ}
CAR	{CAR, DECIDE}, {CAR, WOW}, {CAR, NOW}, {CAR, FINISH}
DECIDE	{DECIDE, WOW}, {DECIDE, HERE}
DIFFERENT	{TOGETHER, DIFFERENT}, {DIFFERENT, MANY}

TABLE 4
Examples of Hand Shapes for Sign Language Spotting: Categories of Hand Shapes Are Described in [1], [4]

Signs	Dominant hand shapes	Non-dominant hand shapes
AND (O)	5>0	D.C.
ARRIVE (T)	B-L	B-L
BIG (T)	bent-L	bent-L
CAR (T)	S	S
DIFFERENT (T)	1	1
FINISH (T)	5	5
O stands for one-handed sign T stands for two-handed sign $D.C.$ means don't care $>$ means that the hand shapes of start and end frames of the signs are changed		

determined by the motion-based sign spotting algorithm. The voting value is calculated by

$$v = \sum_{i=t-t_a/2}^{t+t_a/2} C^g(y_i, L_i), \tag{25}$$

where t is the current frame, t_a is window size (determined by experiment, $t_a = 10$), and $C^g(y_i, L_i)$ is

$$C^g(y_i, L_i) = \begin{cases} 1, & y_i = L_i \\ 0, & \text{otherwise}, \end{cases} \tag{26}$$

where y_i is the label of the observation spotted by the two-layer CRF at position i and L_i is the SVM classification result at position i.

6 Experimental Results and Analysis

6.1 Experimental Environments

The threshold model with CRF formulation is implemented as described in Section 4 by extending the software in [20]. The hand shape verification algorithm is constructed using LIBSVM [8]. Each detected hand region is normalized to a size of 40 × 40 pixels.

For training the threshold model with CRFs and SVMs, 10 sequences for each sign in the 48-sign lexicon were collected. The signer wore colored gloves during collection of the training data; a green glove on the left hand and a purple glove on the right hand. The signer did not wear colored gloves in the test sequences. The start and end points of the ASL signs were labeled manually in the training data and also for the ground truth used for measuring the performance of the proposed method. We captured the video at a rate of 60 frames/s. The original image size was 640 × 480 pixels, which was down sampled to 320 × 240 pixels for the experiments. The ASL data set was captured in a studio environment [1].

A native signer performed 48 signs. Each sign was performed 10 times. Eighteen of 48 signs were one-handed signs and 30 of 48 signs were two-handed signs, as shown in Table 5. Fig. 8 shows six examples of signs used in the experiments. We segmented nonsign patterns in the training data sequences to compare the performance of a conventional CRF in which one class label is a nonsign pattern with the proposed method. As mentioned in Section 1, it is difficult to obtain a representative set of nonsign patterns for training CRFs because motions of nonsign patterns in the training data and test data have large variation. Thus, the labels for nonsign patterns are not correctly modeled to reflect features of the nonsign patterns. We used 522 motion segments for training nonsign patterns.

TABLE 5
Forty-Eight ASL Signs Used in the Experiments

One-handed signs	AND, BLACK, BOY, FARM, FUNNY, GOOD, HOT, KNOW, LIBRARY, LIE, LIKE, MAN, OUT, PAST, SORRY, STRANGE, TELL, YESTERDAY
Two-handed signs	ARRIVE, BICYCLE, BIG, BORN, BUTTER, CAR, COLD, DAY, DECIDE, DIFFERENT, FINISH, HERE, INFORM, INTERPRET, MAGAZINE, MANY, MAYBE, NAME, NIGHT, NOW, RAIN, READ, SHOES, SIT, TAKE-OFF, TOGETHER, WANT, WHAT, WORK, WOW

To measure the accuracy of the proposed method, the word error rate was used [9], [32], [35]. In general, most spotting tasks involve three types of errors; substitution errors, insertion errors, and deletion errors. An insertion error occurs when the spotter reports a nonexistent sign. A deletion error occurs when the spotter fails to spot a sign existing in an input sequence. A substitution error occurs when an input sign is incorrectly classified [1], [9], [39]. The sign error rate (SER) is calculated by

$$\text{SER} = \frac{S + I + D}{N} \times 100, \tag{27}$$

where N is the number of test signs, S is the number of substitution errors, I is the number of insertion errors, and D is the number of deletion errors.

The correct spotting rate is calculated by

$$R = \frac{C}{N} \times 100, \tag{28}$$

where C is the number of correct spottings. The correct spotting, c_s, is defined as

$$c_s = \begin{cases} \text{True}, & \text{if } (\Delta S + \Delta E) < 10 \\ \text{False}, & \text{otherwise}, \end{cases} \tag{29}$$

where $\Delta S = |g_s - s_s|$ and $\Delta E = |g_e - s_e|$. s_s and s_e are the start and end frame numbers of the spotted sign, respectively, and g_s and g_e are the start and end frame numbers of the ground truth of the spotted sign, respectively [25].

Fig. 8. Six examples of ASL signs; S means start and E means end.

TABLE 6
ASL Spotting Results with Continuous Data

Models	N	C	S	I	D	SER	R
DTW	480	302	87	258	91	90.8	62.9
HMM	480	325	78	242	77	82.7	67.7
CRF^f	480	353	63	239	64	76.2	73.5
CRF^g	480	318	53	125	109	59.7	66.2
CRF^{a,f}	480	323	31	253	126	85.4	67.2
LDCRF^a	480	409	33	23	38	19.5	85.2
T-CRF	480	380	65	217	35	66.0	79.1
T-CRF^a	480	368	39	125	73	49.3	76.6
T-CRF^s	480	408	51	250	21	67.0	85.0
T-CRF^{a,s}	480	385	39	84	56	36.5	80.2
Two-layer CRF^{a,s}	480	418	30	10	32	15.0	87.0

a means using the appearance-based sign verification
f means using a fixed threshold without data for non-sign patterns
s means using the shorter sign detector
g means with a label for non-sign patterns in the training data
The window sizes of the CRF and the T-CRF are 2

6.2 Sign Language Spotting with Continuous Data

A DTW, an HMM, a CRF, an LDCRF [24], a T-CRF, and a Two-layer CRF were implemented and compared in an ASL sign spotting application. A continuous DTW was used. A discrete HMM with five states was constructed for each sign. For HMMs and CRFs, a fixed threshold that maximizes the correct spotting rate was selected.

As shown in Table 6, the appearance-based sign verification algorithm decreased insertion and substitution errors. However, it slightly increased deletion errors. As a result, SERs of both the CRF and T-CRF decreased. The short-sign detector increased correct spottings and decreased deletion errors. However, it increased insertion errors. Therefore, both correct spotting rate and SER were increased. However, the increase of SER due to the short-sign detector was compensated for using appearance-based sign verification and subsign reasoning algorithms as shown in the result of the Two-layer CRF^{a,s} model. The correct spotting rate of the T-CRF is higher than that of the CRF^g and the correct spotting rate of the Two-layer CRF^{a,s} is slightly higher than that of the LDCRF. The SER of the LDCRF is slightly higher than that of the Two-layer CRF^{a,s}. This implies that the LDCRF has more errors than the Two-layer CRF^{a,s} for distinguishing in-vocabulary signs and nonsign patterns. See the website http://image.korea.ac.kr/SignLanguageSpotting/ for more detailed experimental video clips.

Fig. 9 shows examples of spotted signs. The spotting result for the sign "MANY" reveals some nonsign patterns

Fig. 9. Six examples of spotted ASL signs; S means start and E means end.

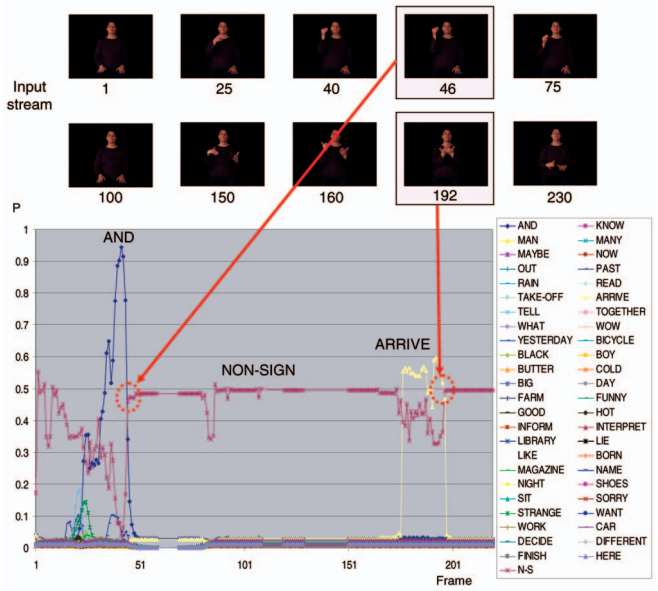

Fig. 10. Temporal evolution of probabilities of signs; circles mark the point from which the probability of a nonsign pattern is higher than those of signs.

at the end of the sign compared with the ground truth of the sign "MANY," as shown in Fig. 8. The direction of movement shown in frames spotted as parts of the sign "MANY" and the hand direction of the sign "MANY" are similar as shown in Figs. 8 and 9. Thus, some nonsign patterns are spotted as parts of the sign "MANY."

Fig. 10 shows a sign spotting result for a sign sequence that contains two in-vocabulary signs and nonsign patterns. The Two-layer CRF^{a,s} was used. The time evolutions of probabilities for in-vocabulary signs and nonsign patterns are illustrated by curves. The label for nonsign patterns has the greatest probability during the first 25 frames. Then, it is followed by the sign "AND." After 46 frames, the probability of the sign "AND" nearly becomes zero and there is a nonsign pattern. After 175 frames, the probability of the sign "ARRIVE" is increased. The start and the end points of in-vocabulary signs and nonsign patterns were obtained by backtracking of the Viterbi path, subsequent to a forward pass as described in Section 3.

Table 7 shows sign language spotting results with the short-sign detector and without the short-sign detector. As mentioned previously, the short-sign detector increased correct spottings and insertion errors. However, it decreased deletion errors.

Table 8 shows the distribution over labels using different window sizes, $W = 0$, 1, and 2. As shown in Table 8, CRFs with longer range dependencies achieves better results. Using a large window in CRFs significantly improves spotting performance.

6.3 Sign Language Recognition with Isolated Data

A classification experiment was performed with isolated signs. In the isolated sign recognition task with an HMM, a CRF, an HCRF, an LDCRF, a T-CRF, and a Two-layer CRF, we used a forward score for each sample to select the model with the highest likelihood. As shown in Table 9, recognition rates of the Two-layer CRF^{a,s} and the T-CRF^{a,s} are higher than those of the HCRF^a, the CRF^a, and the HMM^a.

TABLE 7
ASL Spotting Results of Short Signs with/without the Short-Sign Detector

Signs	T-CRF					T-CRF^s				
	N	C	S	I	D	N	C	S	I	D
AND (O)	10	9	1	2	0	10	9	1	2	0
ARRIVE(T)	10	3	4	0	3	10	9	0	0	1
BICYCLE (T)	10	7	0	1	3	10	8	2	2	0
BIG (T)	10	8	1	0	1	10	7	3	4	0
DECIDE (T)	10	3	7	40	0	10	7	3	59	0
DIFFERENT (T)	10	5	5	0	0	10	9	1	1	0
FARM (O)	10	7	0	2	3	10	9	1	0	0
INFORM (T)	10	9	1	0	0	10	10	0	0	0
FINISH (T)	10	8	2	3	0	10	9	1	3	0
GOOD (O)	10	4	0	0	6	10	4	0	12	6
HOT (O)	10	10	0	12	0	10	10	0	6	0
MANY (T)	10	9	1	0	0	10	9	1	0	0
LIE (O)	10	10	0	1	0	10	8	2	2	0
LIKE (O)	10	10	0	0	0	10	10	0	0	0
MAN (O)	10	6	4	6	0	10	10	0	7	0
NOW (T)	10	0	10	0	0	10	7	3	7	0
OUT (O)	10	9	1	1	0	10	9	1	19	0
PAST (O)	10	10	0	5	0	10	10	0	0	0
SIT (T)	10	10	0	5	0	10	9	1	0	0
STRANGE (O)	10	10	0	0	0	10	9	1	0	0
WANT (T)	10	6	4	0	0	10	9	1	2	0
TELL (T)	10	9	0	2	1	10	9	1	1	0
TOGETHER (T)	10	0	4	0	6	10	2	4	0	4

TABLE 8
ASL Spotting Results with Different Window Size (W)

Models	N	C	S	I	D	SER	R
Two-layer CRF^{a,s}, W=0	480	390	48	16	42	22.0	81.2
Two-layer CRF^{a,s}, W=1	480	410	34	14	36	17.5	85.4
Two-layer CRF^{a,s}, W=2	480	418	30	10	32	15.0	87.0

TABLE 9
ASL Recognition Results with Isolated Data

Models	N	C	S	I	D	SER	R
DTW^a	480	348	132	0	0	27.5	72.5
HMM^a	480	362	118	0	0	24.5	75.4
CRF^f	480	410	70	0	0	14.5	85.4
CRF^{a,g}	480	381	99	0	0	20.6	79.3
HCRF^a	480	436	44	0	0	9.1	90.8
LDCRF^a	480	451	29	0	0	6.0	93.9
T-CRF^{a,s}	480	442	38	0	0	7.9	92.0
Two-layer CRF^{a,s}	480	449	31	0	0	6.4	93.5

The sign recognition rate of the LDCRF^a is slightly higher than that of the Two-layer CRF^{a,s}. The SER of the Two-layer CRF^{a,s} is slightly higher than that of the LDCRF^a.

6.4 Sign Language Spotting with Utterance Data

In this experiment, we spotted signs in video sequences. The signs were performed by a native signer without restrictions. This results in greater difficulty in spotting signs due to the fast signing and large variation in appearance and hand positions of signs [1].

A native signer performed 98 sentences. Each sentence consisted of between three and seven signs. All utterance data consisted of 13,509 frames at a rate of 30 frames/s. Signs were presented in 8,665 frames. The total number of signs is 450; 237 signs were in the vocabulary and 213 signs were not in the vocabulary. The vocabulary consisted of a 48-sign lexicon. A sign that was not in the vocabulary was labeled as a nonsign pattern.

TABLE 10
ASL Spotting Results with Utterance Data

Models	N	C	S	I	D	SER	R
CRF^{f,a}	237	98	70	125	69	111.3	41.3
T-CRF^{a,s}	237	130	48	116	59	94.0	54.8
Two-layer CRF^{a,s}	237	177	21	111	38	72.1	74.6

Fig. 11. Six examples of spotted ASL signs from utterance data; S means start and E means end.

As shown in Table 10, recognition rates of the Two-layer CRF^{a,s} and the T-CRF^{a,s} were higher than that of the CRF^a. The short-sign detector increased correct spottings and decreased deletion errors. It is difficult to correctly detect hand regions in utterance data. Thus, the appearance-based hand verification decreased correct spottings and increased deletion errors.

Fig. 11 shows six examples of spotted signs in a sentence. As shown in Figs. 9 and 11, trajectories of hand movements and the face region differ greatly between training and utterance data sets.

7 Conclusions and Further Research

In this paper, a novel method of designing a threshold model in CRFs has been proposed, which determined an adaptive threshold for distinguishing between signs in the vocabulary and nonsign patterns (which include out-of-vocabulary signs, epentheses, and other movements that do not correspond to signs). The proposed threshold model with CRF is an excellent mechanism for distinguishing in-vocabulary signs and nonsign patterns.

Moreover, a short-sign detector, a hand appearance-based sign verification method, and a subsign reasoning method were included to further improve spotting accuracy. The short-sign detector modeled signs that tend to have slower than the average performance time. The hand appearance-based sign verification algorithm reduced ambiguity among signs that exhibit similar overall hand movements but differ in hand shape. Finally, the subsign reasoning method avoided the premature detection of a sign that is misclassified as part of a long sign.

Experiments demonstrated that our system could detect signs from continuous data with an 87.0 percent spotting rate and recognize signs from isolated data with a 93.5 percent recognition rate versus 73.5 percent and 85.4 percent, respectively, for CRFs without a threshold model, short-sign detection, subsign reasoning, and hand appearance-based sign verification. This paper demonstrated that the proposed threshold model with CRF can accurately detect in-vocabulary signs and nonsign patterns

using an ASL spotting system. Near-term future work includes extending the proposed threshold model with CRFs to HCRFs and improving the sign language spotting performance for the utterance data set.

ACKNOWLEDGMENTS

This research was supported by World Class University Project funded by the Ministry of Education, Science and Technology, Republic of Korea (R31-2008-000-10008-0). This work was also supported by the IT R&D program of MKE/IITA (2008-F-038-01, Development of Context Adaptive Cognition Technology). Stan Sclaroff was supported in part by the US National Science Foundation under Grant 0329009 and Grant 0705749. The authors would like to thank the National Center for Sign Language and Gesture Resources, Boston University for providing the SignStream database.

REFERENCES

[1] J. Alon, "Spatiotemporal Gesture Segmentation," PhD thesis, Computer Science Dept., Boston Univ., 2006.
[2] J. Alon, V. Athitsos, and S. Sclaroff, "Accurate and Efficient Gesture Spotting via Pruning and Subgesture Reasoning," *Proc. IEEE Human Computer Interface Workshop,* pp. 189-198, Oct. 2005.
[3] J. Alon, V. Athitsos, Y. Quan, and S. Sclaroff, "Simultaneous Localization and Recognition of Dynamic Hand Gestures," *Proc. IEEE Workshop Motion and Video Computing,* pp. 254-260, Jan. 2005.
[4] R. Battison, *Lexical Borrowing in American Sign Language.* Linstok Press, 1978.
[5] B. Bauer and K.F. Kraiss, "Video-Based Sign Recognition Using Self-Organizing Subunits," *Proc. 16th Int'l Conf. Pattern Recognition,* pp. 434-437, Aug. 2002.
[6] R. Bowden et al., "A Linguistic Feature Vector for the Visual Interpretation of Sign Language," *Proc. Eighth European Conf. Computer Vision,* pp. 391-401, May 2004.
[7] A. Braffort, "ARGo: An Architecture for Sign Language Recognition and Interpretation," *Proc. Int'l Gesture Workshop Progress in Gestural Interaction,* pp. 17-30, Apr. 1996.
[8] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machine,* http://www.csie.ntu.edu.tw/cjlin/libsvmtools/, 2001.
[9] R.A. Cole et al., *Survey of the State of the Art in Human Language Technology.* Cambridge Univ. Press, 1997.
[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 886-893, June 2005.
[11] T.G. Dietterich, "Machine Learning for Sequential Data: A Review," *Proc. Joint IAPR Int'l Workshops Structural, Syntactic, and Statistical Pattern Recognition,* pp. 15-30, Aug. 2002.
[12] A. Farhadi and D. Forsyth, "Aligning ASL for Statistical Translation Using a Discriminative Word Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1471-1476, June 2006.
[13] A. Forsyth, D. Farhadi, and R. White, "Transfer Learning in Sign Language," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2007.
[14] W. Gao, G. Fang, D. Zhao, and Y. Chen, "Transition Movement Models for Large Vocabulary Continuous Sign Language Recognition," *Proc. Sixth IEEE Int'l Conf. Automatic Face and Gesture Recognition,* pp. 553-558, May 2004.
[15] A. Gunawardana, M. Mahajan, A. Acero, and J.C. Platt, "Hidden Conditional Random Fields for Phone Classification," *Proc. European Conf. Speech Comm. and Technology,* pp. 1117-1120, Sept. 2005.
[16] E.-J. Holden, G. Lee, and R. Owens, "Australian Sign Language Recognition," *Machine Vision and Applications,* vol. 16, no. 5, pp. 312-320, 2005.
[17] K. Imagawa, H. Matsuo, R. Taniguch, and D. Arita, "Recognition of Local Features for Camera-Based Sign Language Recognition System," *Proc. Fourth IEEE Int'l Conf. Automatic Face and Gesture Recognition,* pp. 849-853, Mar. 2000.
[18] J. Jelinek, *Statistical Methods for Speech Recognition.* MIT Press, 1997.
[19] R. Kasturi and R. Jain, *Computer Vision: Principles.* IEEE CS Press, 1991.
[20] T. Kudo, *CRF++: Yet Another CRF Toolkit,* http://chasen.org/taku/software/CRF++/, 2005.
[21] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proc. 18th Int'l Conf. Machine Learning,* pp. 282-289, June 2001.
[22] H.-K. Lee and J.-H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 10, pp. 961-973, Oct. 1999.
[23] A. McCallum, D. Freitag, and F. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation," *Proc. 17th Int'l Conf. Machine Learning,* pp. 591-598, June 2000.
[24] L.-P. Morency, A. Quattoni, and T. Darrell, "Latent-Dynamic Discriminative Models for Continuous Gesture Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, http://sourceforge.net/projects/crf, June 2007.
[25] S. Nayak, S. Sarkar, and B. Loeding, "Unsupervised Modeling of Signs Embedded in Continuous Sentences," *Proc. IEEE Workshop Vision for Human-Computer Interaction,* pp. 81-88, June 2005.
[26] C.W. Ong 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.
[27] *Poser5 Reference Manual,* Poser, Curious Labs, 2004.
[28] A. Quattoni, S. Wang, L.P. Morency, M. Collins, and T. Darrell, "Hidden Conditional Random Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 10, pp. 1848-1852, Oct. 2007.
[29] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, pp. 257-286, 1989.
[30] R.C. Rose, "Discriminant Word Spotting Techniques for Rejection Non-Vocabulary Utterances in Unconstrained Speech," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 105-108, Mar. 1992.
[31] 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.
[32] T. Starner, J. Weaver, and A. Pentland, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 12, pp. 1371-1375, Dec. 1998.
[33] W.C. Stokoe, *Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf, Studies in Linguistics: Occasional Papers 8.* Linstok Press, 1960.
[34] M. Szummer, "Learning Diagram Parts with Hidden Random Fields," *Proc. Eighth Int'l Conf. Document Analysis and Recognition,* pp. 1188-1193, Aug. 2005.
[35] C. Vogler and D. Metaxas, "A Framework for Recognizing the Simultaneous Aspects of American Sign Language," *Computer Vision and Image Understanding,* vol. 81, no. 3, pp. 358-384, 2001.
[36] H.M. Wallach, "Conditional Random Fields: An Introduction," Technical Report MS-CIS-04-21, Univ. of Pennsylvania, 2004.
[37] L.D. Wilcox and M.A. Bush, "Training and Search Algorithms for an Interactive Word Spotting System," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 97-100, Mar. 1992.
[38] D. Xi and S.-W. Lee, "Face Detection and Facial Component Extraction by Wavelet Decomposition and Support Vector Machines," *Proc. Fourth Int'l Conf. Audio- and Video-Based Biometric Person Authentication,* pp. 199-207, June 2003.
[39] H.-D. Yang, A.-Y. Park, and S.-W. Lee, "Gesture Spotting and Recognition for Human-Robot Interaction," *IEEE Trans. Robotics,* vol. 23, no. 2, pp. 256-270, 2007.
[40] H.-D. Yang, S.-W. Lee, and S.-W. Lee, "Multiple Human Detection and Tracking Based on Weighted Temporal Texture Features," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 20, no. 3, pp. 377-391, 2006.
[41] M. Yang, N. Ahuja, and M. Tabb, "Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 8, pp. 1061-1074, Aug. 2002.
[42] R.D. Yang and S. Sarkar, "Detecting Coarticulation in Sign Language Using Conditional Random Fields," *Proc. 18th Int'l Conf. Pattern Recognition,* pp. 108-112, Aug. 2006.

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

Hee-Deok Yang received the BS degree in computer science from Chungnam National University, Daejeon, Korea, in 1998 and the MS and PhD degrees in computer science and engineering from Korea University, Seoul, in 2003 and 2008, respectively. He is currently an assistant professor of the Division of Computer Engineering at Chosun University, Gwangju, Korea. His research interests include sign language recognition, gesture recognition, and face recognition. He is a member of the IEEE.

Stan Sclaroff received the PhD degree from the Massachusetts Institute of Technology in 1995. He is currently a professor of computer science and the chair of the Department of Computer Science at Boston University, where he founded the Image and Video Computing Research Group in 1995. In 1996, he received a US Office of Naval Research (ONR) Young Investigator Award and a US National Science Foundation (NSF) Faculty Early Career Development Award. Since then, he has coauthored numerous scholarly publications in the areas of tracking, video-based analysis of human motion and gesture, deformable shape matching and recognition, as well as image/video database indexing, retrieval, and data mining methods. He has served on the technical program committees of more than 80 computer vision conferences and workshops. He has served as an associate editor for the *IEEE Transactions on Pattern Analysis* (2000-2004 and 2006-present). He is a senior member of the IEEE.

Seong-Whan Lee received the BS degree in computer science and statistics from Seoul National University, Seoul, in 1984 and the MS and PhD degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST) in 1986 and 1989, respectively. From 1989 to 1995, he was an assistant professor in the Department of Computer Science, Chungbuk National University, Cheongju, Korea. In 1995, he joined the faculty of the Department of Computer Science and Engineering, Korea University, Seoul, as an associate professor and is currently a full professor. His research interests include pattern recognition, computer vision, and neural networks. He has more than 250 publications in international journals and conference proceedings and has authored 10 books. He was the recipient of the Annual Best Paper Award of the Korea Information Science Society in 1986, the First Outstanding Young Researcher Award at the Second International Conference on Document Analysis and Recognition in 1993, the First Distinguished Research Award from Chungbuk National University in 1994, and the Outstanding Research Award from the Korea Information Science Society in 1996. He was the founding co-editor-in-chief of the *International Journal of Document Analysis and Recognition* and has been an associate editor of the *Pattern Recognition Journal*, *International Journal of Pattern Recognition and Artificial Intelligence*, and *International Journal of Image and Graphics* since 1997. He has served on the program committees of several well-known international conferences. He is a fellow of the International Association of Pattern Recognition (IAPR), a senior member of the IEEE, and a life member of the Korea Information Science Society.