Pattern Recognition 43 (2010) 2858–2870

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Simultaneous spotting of signs and fingerspellings based on hierarchical conditional random fields and boostmap embeddings

Hee-Deok Yang [a], Seong-Whan Lee [b,c,*]

[a] School of Computer Engineering, Chosun University, Seosuk-dong, Dong-ku, Gwangju 501-759, Republic of Korea
[b] Department of Computer Science and Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-713, Republic of Korea
[c] Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-713, Republic of Korea

ARTICLE INFO

Article history:
Received 19 October 2009
Received in revised form
3 March 2010
Accepted 7 March 2010

Keywords:
Sign language spotting
Fingerspelling spotting
Conditional random field

ABSTRACT

A sign language consists of two types of action; signs and fingerspellings. Signs are dynamic gestures discriminated by continuous hand motions and hand configurations, while fingerspellings are a combination of continuous hand configurations. Sign language spotting is the task of detection and recognition of signs and fingerspellings in a signed utterance. The internal structures of signs and fingerspellings differ significantly. Therefore, it is difficult to spot signs and fingerspellings simultaneously. In this paper, a novel method for spotting signs and fingerspellings is proposed. It can distinguish signs, fingerspellings and non-sign patterns, and is robust to the various sizes, scales and rotations of the signer's hand. This is achieved through a hierarchical framework consisting of three steps: (1) Candidate segments of signs and fingerspellings are discriminated using a two-layer conditional random field (CRF). (2) Hand shapes of segmented signs and fingerspellings are verified using BoostMap embeddings. (3) The motions of fingerspellings are verified in order to distinguish those which have similar hand shapes and different hand motions. Experiments demonstrate that the proposed method can spot signs and fingerspellings from utterance data at rates of 83% and 78%, respectively.

1. Introduction

Sign language is a visual language used by deaf people [1], which consists of two types of action; signs and fingerspellings. Signs are dynamic gestures discriminated by continuous hand motions and hand configurations, while fingerspellings are a combination of continuous hand configurations.

One key task in sign language recognition is detecting signs and fingerspellings in a signed utterance (see Fig. 1). In this paper, signs are words in the predefined vocabulary and fingerspellings are a combination of continuous alphabets e.g., names of people, etc [2,3]. An alphabet is a particular hand configuration [2,3].

The features needed for recognizing sign language are divided into three components; hand shape, hand motion and place of articulation, which is the hand location with respect to the face and body of the signer [4]. In general, fingerspelling spotting is based on the hand location and hand shape rather than the hand motion, and sign spotting is based on three components. The internal structures of signs and fingerspellings differ significantly and thus have different recognition methods [5]. Therefore, it is difficult to recognize signs and fingerspellings simultaneously [6].

For sign recognition, Starner et al. [7], Volger et al. [8], and Yang et al. [9,10], etc., have used Hidden Markov Models (HMMs) and CRFs. In order to distinguish meaningful signs and meaningless (non-sign) patterns, Yang et al. [11] have used a threshold model with CRFs [11,9]. Lee and Kim have also used a threshold model with HMMs [12].

For fingerspelling recognition, Feris et al. [13], Goh and Holden [14], etc., have assumed that candidate fingerspelling segments are detected first. Therefore, they have focused on fingerspelling recognition. Tsechpenakis et al. [6] have used a tracking method and support vector machines (SVMs) to segment fingerspelling. They have focused on fingerspelling segmentation.

In this paper, we will focus on machine vision methods for sign language spotting, i.e., segmentation and recognition of signs and fingerspellings in utterance sentences produced by native signers. The difficulty of sign language spotting stems from the fact that occurrences of signs and fingerspellings vary dynamically in terms of hand motion, hand shape and hand location. The following three problems are considered in order to analyze hand motion and hand location: (1) Signs and fingerspellings can appear within a continuous gesture stream, interspersed with signs, fingerspellings, and non-sign patterns. (2) Fingerspellings can appear in specific regions close to the signer's face [9] and

* Corresponding author at: Department of Computer Science and Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-713, Republic of Korea. Tel.: +82 2 3290 3197; fax.: +82 2 926 2168.
E-mail addresses: heedeok_yang@chosun.ac.kr (H.-D. Yang), swlee@image.korea.ac.kr (S.-W. Lee).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.03.007

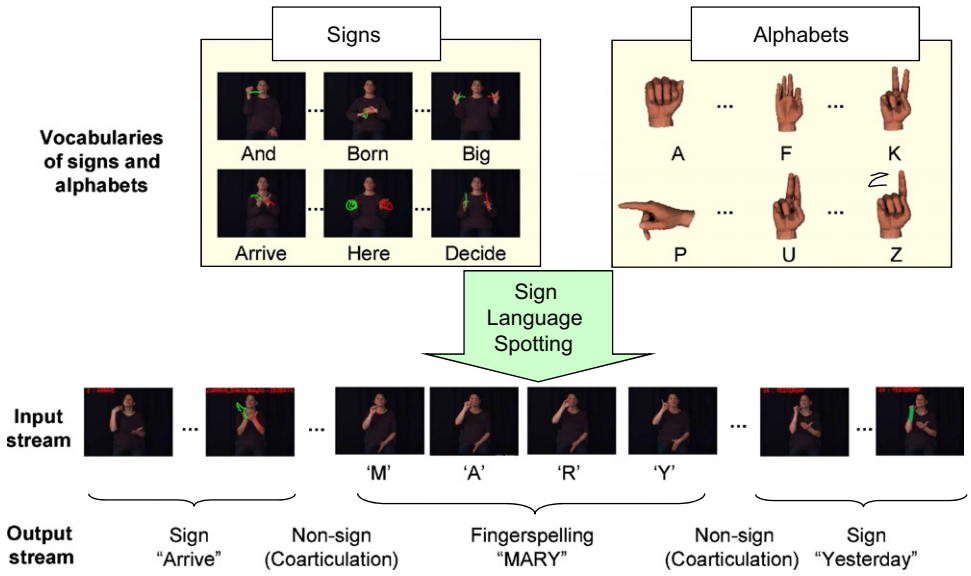

Fig. 1. An example of sign language spotting: Sign language spotting is the task of detecting the beginning and end points of signs and fingerspellings in an input stream and recognizing detected signs and fingerspellings in predefined vocabularies of signs and alphabets.

signs can appear in any region. Thus, it is necessary to consider both the hand motion and hand location simultaneously. (3) Some signs have shared patterns.

In order to solve these problems, a two-layer CRF consisting of a threshold model with CRFs (T-CRFs) [11] and a conventional CRF is applied. The T-CRF, which selects the label with maximum probability, can discriminate signs, fingerspellings and non-sign patterns using both the hand motion and hand location. The conventional CRF can recognize the shared patterns among signs.

In order to recognize the hand shape, previous vision-based methods have used nearest neighbor template matching, deterministic boosting methods [15,16] and shape descriptor methods [13], etc. In this paper, BoostMap embeddings, which are robust to various scales, rotations and sizes of the signer's hand, are applied in order to recognize the hand shape.

The proposed sign and fingerspelling spotting method consists of three steps: (1) Candidate segments of signs and fingerspellings are discriminated using a two-layer CRF. (2) The hand shapes of the detected candidate segments are verified using BoostMap embeddings. (3) The motions of fingerspellings are verified in order to distinguish fingerspellings which have similar hand shapes and different hand trajectories.

The rest of this paper is organized as follows: Section 2 reviews related work and is divided into three categories; sign recognition, fingerspelling recognition and hand shape recognition. Section 3 provides the proposed hierarchical sign and fingerspelling spotting method. Section 4 presents the sign and fingerspelling spotting system. Section 5 presents the experimental results and discusses their implications. Section 6 concludes this paper.

2. Related work

The representative sign language systems and methods are briefly reviewed in this section. For a comprehensive review of sign language analysis, refer to the survey paper [17]. Also, for shape analysis with nearest neighbor methods, refer to papers [15,18].

2.1. Methods for sign recognition

Dynamic time warping (DTW) [19], HMMs [7,8], and CRFs [11,9], etc., have been used to recognize and spot signs.

Starner et al. [7] presented an American sign language (ASL) recognition system based on HMMs. The experiments involved two systems and 40 signs. The first system used a camera on a desk to observe the signer, and had an accuracy of 92%. The second system used a camera on the signer's cap to observe the signer, and had an accuracy of 98%.

Yang and Sarkar [9] proposed an ASL spotting system with CRFs. They extracted key frames from sentences and then labeled each one as either coarticulation (non-sign) or sign label. They modeled coarticulation using training data. However, it is difficult to obtain a set of coarticulation patterns for model training, because there are a wide range of possible patterns.

Ding and Martinez [4] proposed an ASL recognition method to model the hand shape, hand motion and hand location, which are considered the three basic components needed for sign language recognition. They reconstituted 3D hand shapes and obtained the hand trajectories. Experiments were performed on a database of isolated ASL signs using a tree-based classifier.

Yang et al. [10] proposed an ASL recognition method based on an enhanced level building algorithm. They used a dynamic programming approach to spot signs without modeling coarticulation patterns. They used hand motion for recognizing signs.

Lichtenauer et al. [19] proposed a hybrid approach, which used statistical DTW for time warping and independent classification methods on the wrapped features for sign recognition. The method used discriminative features (DFs). As a result, non-DFs were discarded in order to reduce noise.

Yang et al. [11] proposed an adaptive threshold model based on a CRF in order to distinguish signs and non-sign patterns. A conventional CRF was initially constructed without a non-sign pattern label. Then, an adaptive threshold model with a CRF was constructed by adding a non-sign pattern label into the CRF using the weights of its state and transition feature functions. They did not use training data for non-sign patterns.

2.2. Methods for fingerspelling recognition

It is difficult to detect hands in arbitrary environments. In order to solve this problem, Goh and Holden, Feris et al. [6,13,14] have assumed that the hand is restricted to facing the camera in a

uniform background environment or the camera focuses on the hand region or the input stream consists of isolated data [17].

Goh and Holden [14] proposed a fingerspelling recognizer for Australian sign language (Auslan). The system could recognize Auslan alphabets from video sequences. A combination of geometric and motion features was extracted using the optical flow. The sequence of feature sets was classified with HMMs. Two experiments were performed with 20 fingerspellings on the word and letter levels. The recognition rate of the latter was higher than that of the former.

Feris et al. [13] proposed a fingerspelling recognition system which could discriminate complex hand configurations. They used a multi-flash camera having four flashes to extract hand shapes. They extracted a depth image and captured several images using different lighting sources. The scale-invariant feature transform (SIFT) was used to extract features.

Tsechpenakis et al. [6] proposed a hand tracking method to detect signs and fingerspellings. The method used discrete and continuous trackers to reduce accumulated tracking errors. They discriminated signs and fingerspellings with SVMs.

Laura et al. [16] proposed a fingerspelling recognition method based on collaboration between feature extraction and classification. Two tasks were performed in parallel and the results were independent. They used a multi-HMM framework to support collaboration between feature extraction and classification.

Liwicki and Everingham [1] proposed a fingerspelling recognition method using a boostrapping method to segment the hands, a histogram of oriented gradients descriptor to represent extracted features, and logistic regression to classify the hand shapes. They also used HMMs in order to recognize continuous letters on the word level.

2.3. Methods for shape recognition

As mentioned in Section 1, fingerspellings are a combination of continuous alphabets which mimic the letters of the native spoken language and are represented by particular hand configurations [2].

Embedding and boosting methods have been used to retrieve and classify objects in a large database [15,18,20–23]. More recently, nearest neighbor retrieval based on AdaBoost has been researched [15]. Nearest neighbor retrieval can be divided into three categories: (1) measuring the distance in non-metric space e.g., Chamfer distance [24], shape context matching [20], and DTW [21], (2) measuring the distance in metric space e.g., Lipschitz embeddings and SparseMap [18], and (3) measuring the distance in metric and non-metric spaces e.g., BoostMap embeddings [15].

Bourgain [25] proposed Lipschitz embeddings which assume that two nearby points are closer to each other than to any other point. In general, this assumption does not hold for non-metric distance measurement. In order to solve this problem, embedding methods based on non-metric distance measurement have been researched.

Belongie et al. [20] proposed shape context descriptors to compute the distances between shapes. Shape context descriptors provide correspondences between shapes using graph matching, which measures the distances between shapes when different numbers of features are detected on them. However this method cannot handle scale changes of objects.

Lowe [26] proposed a method for feature generation called SIFT. The SIFT features use local information and are extracted using the object shape at particular points.

Mori et al. [22] proposed generalized shape contexts (GSCs), which is an extension of shape contexts using local tangent information at particular points. The GSC aggregates edge orientations into a histogram as SIFT. GSC considers global information, while SIFT represents local information.

Athitsos et al. [15] proposed BoostMap, which is a method for efficient nearest neighbor retrieval. Database and input objects are embedded into a vector space and each embedding is treated as a classifier. They reduced the problem of embedding construction by combining many weak classifiers into a strong classifier.

3. Proposed sign and fingerspelling spotting method

In order to spot signs and fingerspellings in a signed utterance, we use a hierarchical framework. First, a two-layer CRF is applied to discriminate candidate segments of signs and fingerspellings. After detecting the candidate segments, their hand shapes are verified using BoostMap embeddings [15]. Finally, the motions of fingerspellings are verified with the CRF.

3.1. Hierarchical framework for sign and fingerspelling spotting

As mentioned in Section 1, the hand shape, hand motion and hand location are the basic three components used to recognize sign language. It is difficult for a sign language recognition system to process these three components simultaneously. In order to combine the three components for a sign language recognition system, a hierarchical framework based on CRFs and BoostMap embeddings is proposed. As shown in Fig. 2, the proposed sign and fingerspelling spotting method consists of three steps: (1) Candidate segments of signs and fingerspelling are discriminated via a two-layer CRF. (2) The hand shapes of the candidate segments are verified using BoostMap embeddings. (3) The motions of the fingerspellings are verified with a conventional CRF.

3.2. Two-layer CRF for sign and fingerspelling segmentation

There are ambiguities between signs that have similar hand movements and different hand shapes, as shown in Fig. 3. The two-layer CRF architecture is applied to deal with this problem. The first layer uses a T-CRF and the second layer uses a conventional CRF. In the first layer, the T-CRF discriminates signs, fingerspellings and non-sign patterns. Then, in the second layer, shared patterns among signs are recognized using the segmented sign sequence from the first layer.

Starner et al. [7], Volger and Metaxas [8], etc., have used a fixed threshold model to recognize signs. However, it is difficult to select a fixed threshold that is effective for all labels [11].

In order to discriminate signs and non-sign movements, Yang et al. [11] proposed an adaptive T-CRF. An adaptive threshold model with a CRF is applied in order to automatically detect signs and fingerspelling segments in a signed utterance. The threshold model is able to distinguish signs, fingerspellings and non-sign patterns without training data for non-sign movements.

In a CRF, the probability of a label sequence $\mathbf{y}$ given an observation sequence $\mathbf{x}$ is found using a normalized product of potential functions. Each product of potential functions is represented by [11,27,28]

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

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 the labels of observation sequence $\mathbf{x}$ at positions i and $i-1$, and λ_v and μ_m are the weights of the transition and state feature functions, respectively.

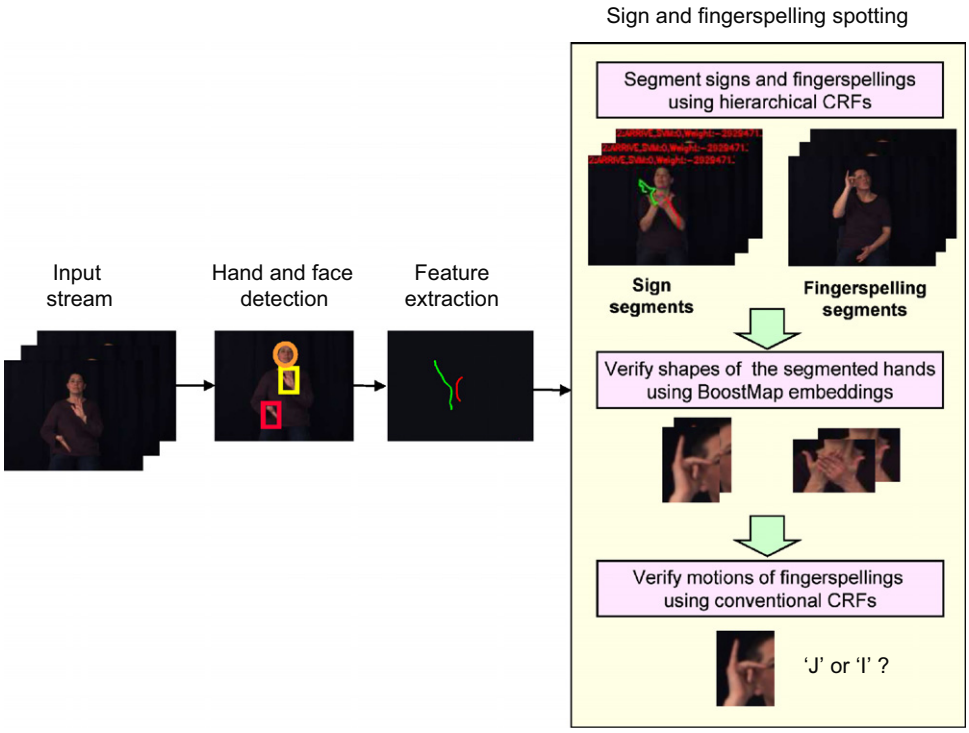

Fig. 2. Overview of the proposed method for spotting sign and fingerspelling.

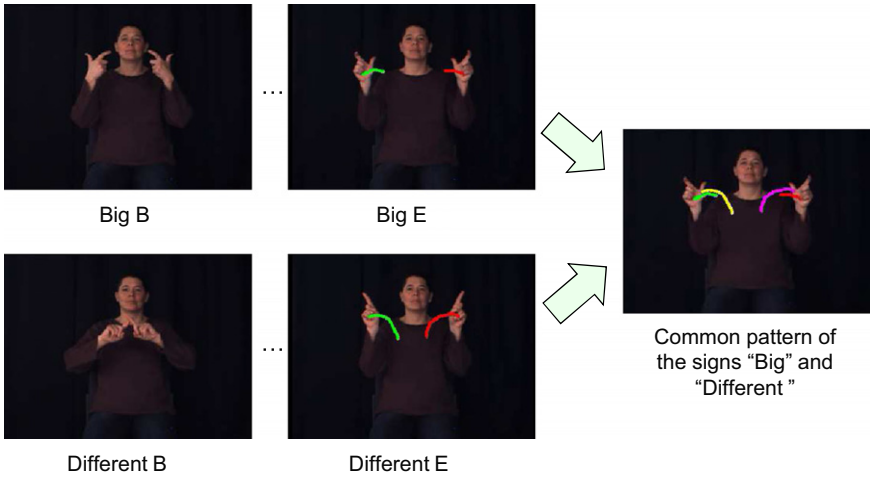

Fig. 3. Common pattern of the signs "big" and "different". B and E means the beginning and end points of signs, respectively.

A state feature function represents whether or not a feature is observed at a particular state (sign or fingerspelling). A transition feature function represents whether or not a feature is observed between two states.

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

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\theta(\mathbf{x})}\exp\left(\sum_{i=1} F_\theta(y_{i-1},y_i,\mathbf{x},i)\right), \tag{2}$$

where θ is weight of feature functions, $F_\theta(y_{i-1},y_i,\mathbf{x},i) = \sum_v \lambda_v t_v(y_{i-1},y_i,\mathbf{x},i) + \sum_m \mu_m s_m(y_i,\mathbf{x},i)$, and $Z_\theta(\mathbf{x})$ is the normalization factor

$$Z_\theta(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left(\sum_{i=1} F_\theta(y_{i-1},y_i,\mathbf{x},i)\right). \tag{3}$$

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 [27,28].

First, a conventional CRF which has labels $S=\{Y_1,\ldots,Y_l,Y_{l+1}\}$ is built, where Y_1 to Y_l are the sign labels and l is the number of signs (the words in the predefined vocabulary). All fingerspellings are represented using a single label, Y_{l+1}. A T-CRF is constructed by adding the non-sign pattern label G into the conventional CRF in order to discriminate signs, fingerspellings and non-sign patterns. Therefore, the T-CRF has labels $S_T=\{Y_1,\ldots,Y_l,Y_{l+1},G\}$.

As mentioned in [11], the weight of the state feature function of the state for non-sign patterns G is calculated by applying Dugad et al.'s method;

$$\mu_m(G) = \overline{\mu}_m + T_d\sqrt{\sigma_{\mu_m}}, \tag{4}$$

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

In sign language sentences, non-sign movements appear more frequently than signs and fingerspellings. Therefore, the weight of the self-transition feature function of the non-sign pattern label G is calculated by

$$\lambda_v(G,G) = \underset{k=1,\ldots,l+1}{\operatorname{argmax}} \lambda_v(Y_k,Y_k) + \kappa, \tag{5}$$

where κ is the weight of the self-transition feature function of the non-sign pattern label G [11].

The weights of transition feature functions from other labels to the non-sign movement label G are assigned by

$$\forall_{k \in \{1,\ldots,l+1\}} \lambda_v(Y_k,G) = \frac{\lambda_v(Y_k,Y_k)}{l+1}. \tag{6}$$

The weights of transition feature functions from the non-sign movement label G to other labels are assigned by

$$\forall_{k \in \{1,\ldots,l+1\}} \lambda_v(G,Y_k) = \frac{\lambda_v(G,G)}{l+1}. \tag{7}$$

In order to detect shared patterns among signs, T-CRF is applied first. The second layer CRF which is a conventional one, models the shared patterns. An input observation of the second layer CRF $\mathbf{x}'$ is a sequence of distinct signs, which is detected by the T-CRF, as shown in Fig. 4. As a result, the fist layer CRF has labels $S_T=\{Y_1,\ldots,Y_l, Y_{l+1},G\}$ and the second layer CRF has labels $S_C=\{Y_1,\ldots,Y_l,Y_{l+1}\}$.

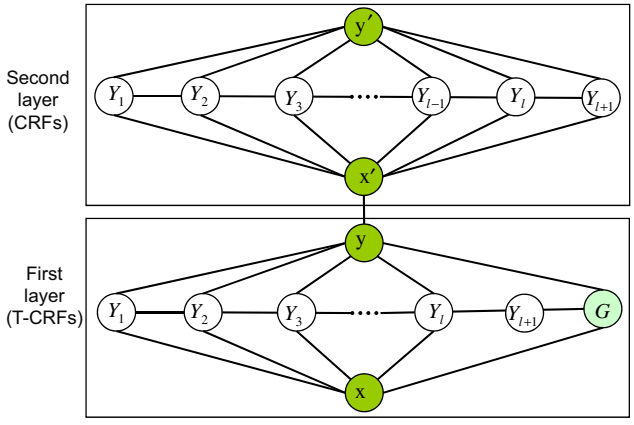

Fig. 4. Structure of the two-layer CRF. Y_1 to Y_l are the sign labels, l is the number of signs (the words in the predefined vocabulary), Y_{l+1} is the fingerspelling label, and G is the non-sign pattern label. $\mathbf{x}$, $\mathbf{x}'$ are observation sequences and $\mathbf{y}$, $\mathbf{y}'$ are the labels of the observation sequences $\mathbf{x}$, $\mathbf{x}'$, respectively.

As shown in Fig. 5, part of the sign "Decide" is confused with the sign "Here" in the region of the ground truth [11]. A sequence of distinct signs $\mathbf{x}'$, which is discriminated by the first layer CRF, is an input observation of the second layer CRF. The input observations of the second layer CRF are "Decide" and "Here", which are spotted in the first layer CRF, as shown in Fig. 5. Table 1 shows examples of the extracted subsign patterns in the training data described in Section 5. The second layer CRF is trained using the detected subsign patterns in the training data. The detail algorithm is described in [11].

3.3. BoostMap embedding for hand shape recognition

The two-layer CRFs described in Section 3.2 discriminates candidate signs, fingerspellings and non-sign patterns, using both hand motion and hand location as features. The BoostMap method, which is an embedding method, is applied in order to recognize hand shapes in candidate sign and fingerspelling segments.

Nearest neighbor classification is the task of identifying objects in the database which are the most similar to an input object. In order to measure the distance between an object in the database and an input object, embedding and boosting methods, etc., have been applied [15,29]. A common approach to solve this problem is the use of embedding methods, which embed objects into a low-dimensional space and measure the distances between them.

Table 1
Examples of subsign patterns extracted from the training data.

Supersigns	Subsigns ($\mathbf{x}'$)
AND	{TELL, AND}, {PAST, AND}
DECIDE	{DECIDE, WOW}, {DECIDE, HERE}
DIFFERENT	{TOGETHER, DIFFERENT}, {DIFFERENT, MANY}
FINISH	{FINISH, MANY}, {FINISH, CAR}
HERE	{HERE, MANY}, {HERE, WOW}

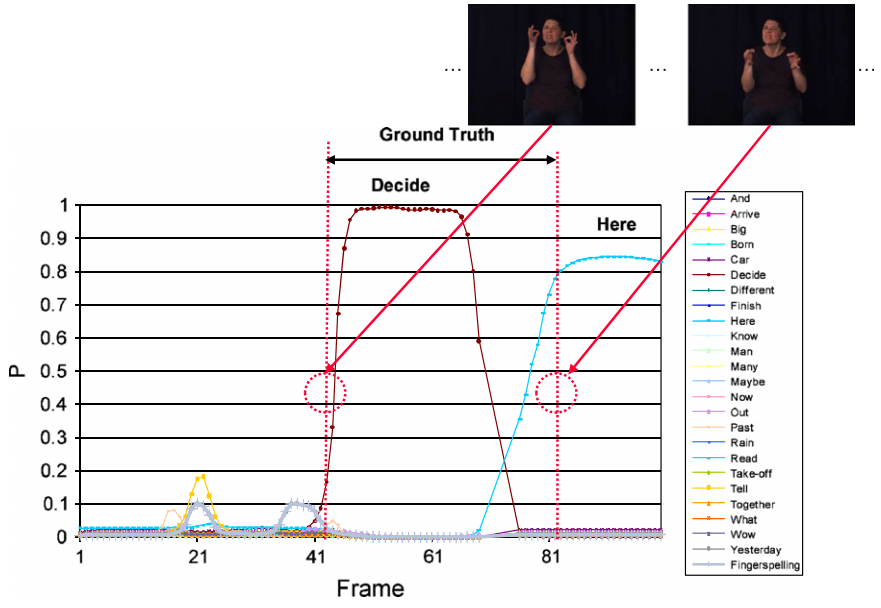

Fig. 5. Temporal evolution of probabilities of signs including subsign patterns: the vertical dashed line indicates the ground truth of the sign "decide".

Given an arbitrary space, $\mathbb{X}$, and a distance measure, D, which represents the distance between objects in $\mathbb{X}$, the distance measure, D, is extended in order to calculate the distance between an object in $\mathbb{X}$ and an object in a subset of $\mathbb{P}$, by [15]

$$D(X,\mathbb{P}) = \min_{P \in \mathbb{P}} D(X,P), \tag{8}$$

where $X \in \mathbb{X}$, $\mathbb{P} \subset \mathbb{X}$, and $P \in \mathbb{P}$.

A simple one-dimensional embedding, $F^{\mathbb{P}}$, is a function that maps $\mathbb{X}$ space to $\mathbb{P}$ space given by

$$F^{\mathbb{P}}(X) = D(X,\mathbb{P}). \tag{9}$$

The set $\mathbb{P}$ is called a reference set, if $\mathbb{P}$ is composed of a single object, which is called a reference object. Eq. (9) is rewritten as

$$F^{P}(X) = D(X,P). \tag{10}$$

A multi-dimensional embedding is constructed by combining one-dimensional embeddings. A d-dimensional embedding, which is a function that maps $\mathbb{X}$ to $\mathbb{P}^d$ space, is represented by $F(X)=(F_1(X),\ldots,F_d(X))$.

In BoostMap, an embedding F is proximity-preserving, when it satisfies the below condition, for all objects, X, A, and $B \in \mathbb{X}$

$$D(X,A) \le D(X,B) \Longleftrightarrow \Delta_F(X,A) \le \Delta_F(X,B), \tag{11}$$

where Δ_F is a distance measure in $\mathbb{P}^d$ space.

In order to convert an embedding F to a classifier $\tilde{F}$, Eq. (11) is modified as [15]

$$\tilde{F}(X,A,B) = \mathrm{sign}(\Delta_F(X,A) - \Delta_F(X,B)), \tag{12}$$

where sign(v) is set to 1 when $v > 0$, 0 when $v = 0$, and -1 when $v < 0$.

We can estimate whether X is closer to A or B by checking the right term of Eq. (12) [15]. Many binary classifiers can be extracted by applying Eq. (12) to each one-dimensional embedding. Each embedding F maps $(\mathbb{X},D)$ to $(\mathbb{P}^d,\Delta)$, where D and Δ are the distance measure in spaces $\mathbb{X}$ and $\mathbb{P}^d$, respectively.

The input of AdaBoost is a set of randomly selected one-dimensional embedding and training triples (X, A, B) of objects [15]. The output of AdaBoost is a classifier, H, which can be described by

$$H(X,A,B) = \sum_{j=1}^{d} \alpha_j \tilde{F}_j(X,A,B), \tag{13}$$

where $\tilde{F}_j(X,A,B)$ is a weak classifier which is described in Eq. (12) and α_j is a corresponding weight [15].

According to $H(\cdot)$ as described in Eq. (13), a high-dimensional embedding F_{out} is defined by

$$F_{out}(X) = (F_1(X),\ldots,F_d(X)). \tag{14}$$

We can obtain Eq. (15) according to Eqs. (13) and (14)

$$\Delta(F_{out}(X),F_{out}(Y)) = \sum_{j=1}^{d} \alpha_j |F_j(X) - F_j(Y)|. \tag{15}$$

Eqs. (13)–(15) are proved in [30].

4. Sign and fingerspelling spotting system

We provide details of the vision-based sign and fingerspelling spotting system that is used for testing the proposed method. In order to spot signs and fingerspellings in a signed utterance, a hierarchical framework consisting of three steps is proposed. Two two-layer CRFs are constructed to spot signs and fingerspellings: (1) one-handed signs and fingerspellings and (2) two-handed signs. All the fingerspellings are one-handed movements, thus, the movements of the non-dominant hand are not considered in our system. The beginning and end points of all signs and fingerspellings are obtained by back-tracking of the Viterbi path subsequent to a forward pass [11,27,28]. Fig. 6 shows the flowchart of the proposed sign and fingerspelling spotting system.

4.1. Hand and face detection and tracking

In order to spot signs and fingerspellings in a signed sentence, we first detect the face and hands of the signer. Video sequences are recorded and one color camera shows a front view of the upper body of the signer in a studio environment [31]. The signer wears dark clothes on a uniform background, as shown in Fig. 11.

An Adaboost-based face detector [32] is used to detect the signer's face. Once the face is detected, the mean and covariance of the face skin pixels in rg color space is extracted. Then, in order to detect the signer's hands, skin color and motion cues are combined. A frame differencing method is used to detect hand motion [11,33].

An appearance-based hand tracking algorithm is used to track the hand region [34]. If the hand detector successfully extracts the hand region, then the hand appearance is stored as a template [11].

4.2. Feature extraction

Both motion- and location-based features in 2D space are used to train the model. Eight features are extracted using the detected hand and face regions, as shown in Table 2 [6,11].

The feature, P_{LH}, represents the location of the left hand with respect to the signer's face. The distance between the face and left hand, d_{FLH}, and the angle between them, θ_{FLH}, are extracted. Then, the feature vector $\{\theta_{FLH},d_{FLH}\}$ is clustered into an index using an EM-based Gaussian mixture model (GMM) [11]. Fig. 7 shows the two extracted features, d_{FLH} and θ_{FLH}. From this plot, it can be seen that fingerspellings have less differences between successive hand positions than signs [6].

The hand symmetry, S_{TH}, is calculated from the distance between the two hand locations and the hand occlusion, O_{TH}, is determined from the ratio of the overlapped region of the two hands [11].

The directional codewords, C_{LH} and C_{RH} represent the hands' moving direction and are one of eight direction codewords or one dummy codeword. The dummy codeword represents the case where there is negligible movement between two positions [12,11].

The differences, D_L and D_R, which are calculated using the L 2 norm, represent the displacements obtained from successive hand positions [6].

4.3. Motion- and location-based sign and fingerspelling segmentation

The main goal of motion- and location-based sign and fingerspelling segmentation is to classify a signed utterance into sign, fingerspelling and non-sign segments. In order to solve this problem, the two-layer CRF described in Section 3 is applied.

First, the conventional CRF is trained with the eight extracted features. Second, the T-CRF [11] is built using the weights of the state and transition feature functions of the CRF, by augmenting it with one additional label that can play the role of an adaptive threshold. Then, the two-layer CRF is built, consisting of a T-CRF and a conventional CRF, as described in Section 3.2.

4.4. Shape-based sign and fingerspelling verification

The two-layer CRF is useful for representing information about the hand trajectory and hand location. However, there are

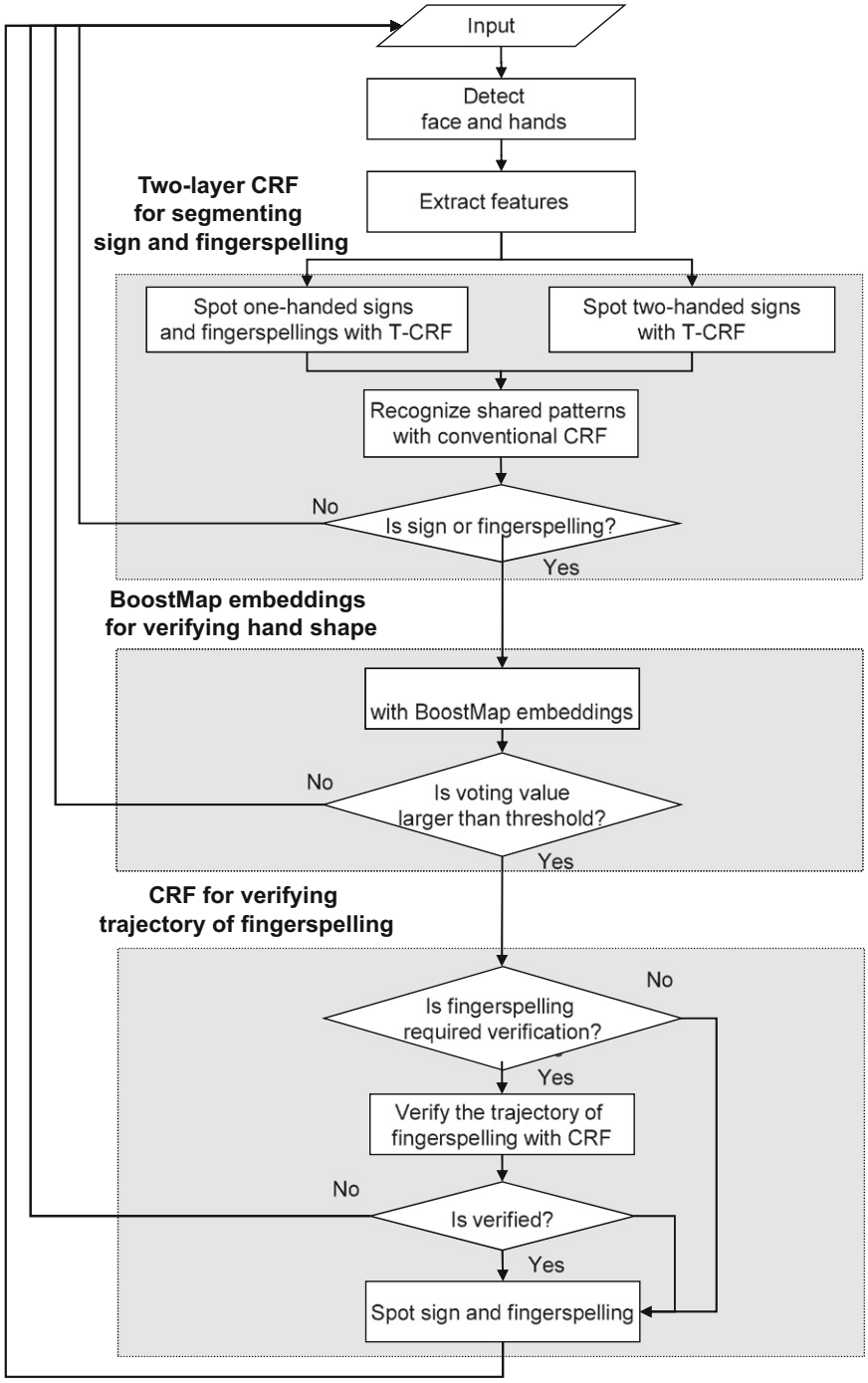

Fig. 6. Flowchart of the proposed sign and fingerspelling spotting system.

ambiguities between signs that have similar hand movements and different hand shapes. In addition, all fingerspellings are represented with a single label, as mentioned in Section 3.2. Therefore, the hand shapes of the detected candidate segments of signs and fingerspellings are recognized using BoostMap embeddings.

The main goal of the hand shape-based sign and fingerspelling verification method is two-fold: (1) Deciding whether or not a sign segment spotted via a motion- and location-based spotting method is accepted as a sign. (2) Identifying the alphabets in a fingerspelling segment.

In order to recognize the hand shape, BoostMap embeddings are applied. This method is robust to various scales, rotations and sizes of the signer's hands. Synthetic hand images were generated using Poser 7 [35] to train the model. Each sign begins and ends with a specific hand shape and each alphabet has unique hand shapes [36,11]. Fig. 8 shows the ASL hand shape letters used for training BoostMap in the experiments. As shown in Fig. 9, the letters 'K' and '2' are very similar. Thus, in our system, a database with 17 ASL letters consists of unique hand shapes. For each hand shape, 864 images are generated.

The Chamber distance with edge images is used to compare images. Edge images are automatically generated by Poser for the training data and are extracted using a Canny edge detector for testing data from the hand region detected in an input image.

For verifying signs and fingerspellings, the hand appearance is verified over a period of several frames. Then, voting is used to decide whether to accept or reject the sign and fingerspelling [11].

4.5. Motion-based alphabet verification

The letters 'I', 'J' and 'Z', '1' have similar hand shapes and this can lead to ambiguities in fingerspelling spotting, as shown in Fig. 10. There are ambiguities between alphabets that exhibit similar hand shapes and different hand motions. The hand motions of alphabets are verified in order to distinguish alphabets which have similar hand shapes and different hand trajectories. In order to verify the hand motions of alphabets, a conventional CRF is applied.

5. Experimental results and analysis

5.1. Experimental environments

For the training models used in each step, two datasets have been collected. A native signer performed 98 sentences, each consisting of 3–10 signs and fingerspellings. Table 3 shows examples of signed utterance data. Fig. 11 shows the signed sentence, "JOHN arrived yesterday".

The signer wore green and purple gloves on the left and right hand, respectively, during collection of the training data. The signer did not wear colored gloves in the test sequences. The signer did not wear colored gloves in the test sequences. The video sequences had resolutions of 640×480 pixels, and were recorded at 60 frames/s. The videos were down sampled to 320×240 pixels for the experiments. The ASL data set was captured in a studio environment [11,31].

There were a total of 515 and 50 signs and fingerspellings, respectively, for the entire set of utterance data. Some signs appeared once or twice in the entire set of utterance data, thus, they were treated as non-sign patterns. The sign vocabulary

Table 2
Eight motion- and location-based features [6,11].

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
D_L	Difference between the current and the previous positions of the left hand
D_R	Difference between the current and the previous positions of the right hand

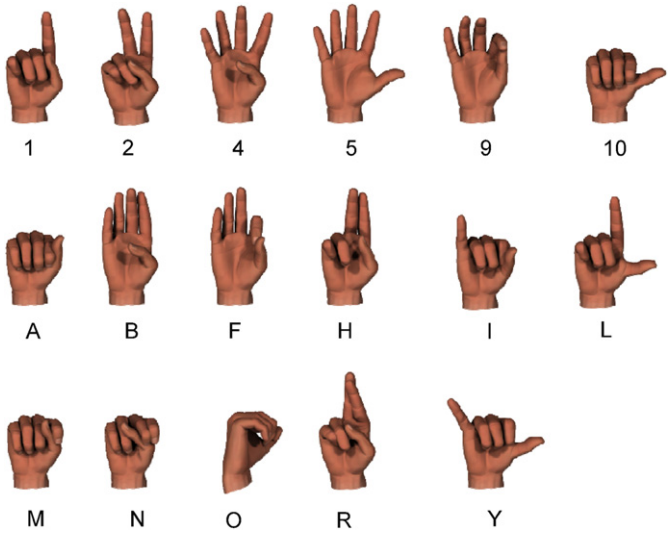

Fig. 8. Examples of ASL letters used for training the BoostMap embeddings.

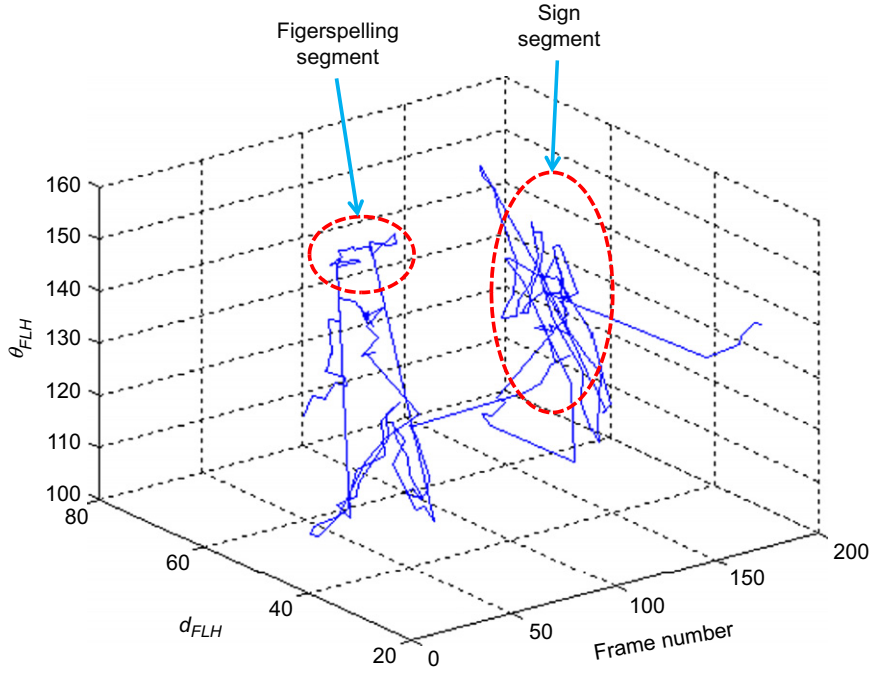

Fig. 7. Plot of the two extracted features, the distance, d_{FLH}, and the angle, θ_{FLH}, between the face and left hand.

consisted of 24 signs, each of which appeared at least five times in the entire set of utterance data. Of the total of 24 signs, 7 were one-handed signs and 17 were two-handed signs, as shown in Table 4. The fingerspelling vocabulary for the 17 ASL letters consisted of unique hand shapes, as shown in Fig. 8. The 222 signs and 50 fingerspellings were in the vocabulary and 293 signs were not, for the entire set of utterance data. A sign which was not in the vocabulary was labeled as a non-sign pattern. In other words, the proposed system can spot the 24 signs and 17 ASL letters in the sign and fingerspelling vocabularies, respectively. Table 5 shows a summary of the database used in the experiments.

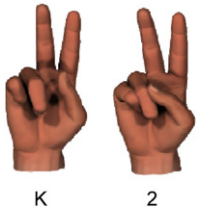

Fig. 9. An example of similar hand shapes: The letters 'K' and '2'.

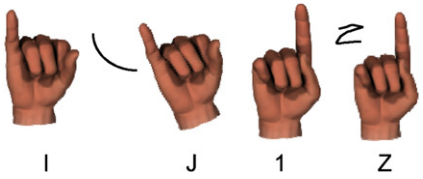

Fig. 10. An example of the letters 'I', 'J' and 'Z', '1' which have similar hand shapes and different hand trajectories.

Table 3
Examples of experimental utterance data.

Videos	Signed utterance	Meaning
1	JOHN (F) arrive (S) yesterday (S).	JOHN arrived yesterday.
2	JOHN (F) decide (S) what (S) yesterday (S).	What did JOHN decide yesterday?
3	MARY (F) know (S) rain (S) yesterday (S) here (S).	MARY knew that it rained here yesterday.

S stands for sign
F stands for fingerspelling

As with many other pattern recognition approaches using statistical models, the database is inadequate for reliable estimation of CRF parameters. In order to solve this problem, we have used cross-validation to measure the performance of the proposed method. We also tried to alleviate the problem by synthesizing sign variations via the addition of Gaussian noise to the sign trajectories.

For training the BoostMap embeddings, an ASL hand shape dataset consisting of 12,960 synthetic images of hand shapes was generated, as mentioned in Section 4.4.

In order to measure the accuracy of the proposed method, the word error rate was used [7,8,11]. In general, most spotting tasks involve three types of errors, namely, substitution, insertion and deletion errors [11]. The sign error rate (SER) is calculated by

$$SER = \frac{S+I+D}{N} \times 100, \tag{16}$$

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

The correct detection rate is also calculated by

$$R = \frac{C}{N} \times 100, \tag{17}$$

where C is the number of correct detections.

The HMM and conventional CRF were implemented and compared in the ASL spotting application. A discrete HMM with five states was constructed for spotting signs and fingerspellings. For the HMM and CRF, a fixed threshold that maximizes the correct detection rate was selected. Also, the BoostMap embeddings and SVM were implemented and compared in order to verify the hand shapes.

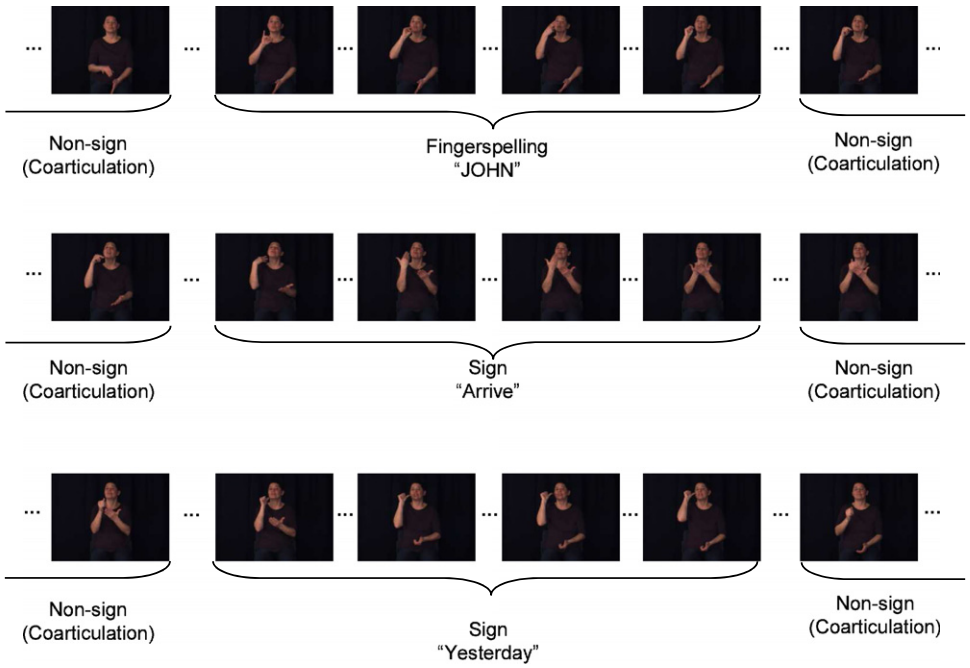

Fig. 11. An example of a signed utterance used for training and testing the proposed method. The signed utterance is "JOHN arrived yesterday".

Table 4
24 ASL signs used in the vocabulary.

One-handed signs	And, Know, Man, Out, Past, Tell, Yesterday
Two-handed signs	Arrive, Big, Born, Car, Decide, Different, Finish, Here, Many, Maybe, Now, Rain, Read, Take-off, Together, What, Wow

Table 5
Database used in the experiments.

Category	Number of signs or fingerspellings
Number of videos	98
Number of signs in utterance sentences	515
Number of fingerspellings in utterance sentences	50
Vocabulary size of signs	24
Vocabulary size of fingerspellings	17
Number of signs in utterance sentences which are in the vocabulary	222
Number of fingerspellings in utterance sentences which are in the vocabulary	50

Table 6
ASL sign spotting results with utterance data.

Models	*N*	*C*	*S*	*I*	*D*	*SER*	*R*
HMM^{BME}	222	93	70	125	64	111.6	41.8
CRF	222	101	65	114	61	108.1	45.4
CRF^{SVM}	222	113	58	101	56	96.8	50.9
CRF^{BME}	222	120	53	115	54	100.0	54.0
Proposed $method^{SVM}$	222	173	20	111	34	74.3	77.9
Proposed $method^{BME}$	222	185	18	99	24	63.5	83.3

SVM means using SVM based hand shape verification.
BME means using BoostMap embeddings based hand shape verification.

Table 7
ASL fingerspelling spotting results with utterance data.

Models	*N*	*C*	*S*	*I*	*D*	*SER*	*R*
HMM^{BME}	50	13	15	45	22	164.0	26.0
CRF	50	15	14	30	21	130.0	30.0
CRF^{SVM}	50	20	12	22	18	104.0	40.0
CRF^{BME}	50	23	10	24	17	102.0	46.0
Proposed $method^{SVM}$	50	35	8	19	7	68.0	70.0
Proposed $method^{BME}$	50	39	6	15	5	52.0	78.0

5.2. Sign and fingerspelling spotting with utterance data

As shown in Tables 6 and 7, the sign and fingerspelling spotting rates of the proposed method with BoostMap embeddings were higher than those of the other methods. The proposed method increased the correct detection rate and decreased insertion and substitution errors by using a hierarchical framework for discriminating signs, fingerspellings and non-sign patterns. The BoostMap embedding method decreased insertion and substitution errors by verifying the hand shape, but it reduced the correct detection rate, because it has its own classification errors. However, the reduced correct detection rate was compensated by the insertion and substitution errors. The recognition rates of fingerspellings were evaluated using the word level.

Fig. 12 shows an example of the sign and fingerspelling spotting results for the signed utterance "MARY arrived yesterday". The time evolution of the probabilities of in-vocabulary signs and fingerspellings and out-vocabulary non-sign patterns is illustrated by the curves. The non-sign pattern label has the greatest probability during the first 23 frames and this is followed by the fingerspelling label. After 55 frames, the probability of a fingerspelling approaches zero and a non-sign pattern appears. After 65 frames the probability of the sign "Arrive" is increased. The start and end points of in-vocabulary signs and fingerspellings and out-vocabulary non-sign patterns were obtained by back-tracking of the Viterbi path, subsequent to a forward pass. Fig. 13 shows the fingerspelling verification results with the BoostMap embeddings in the fingerspelling segment of Fig. 12. "MARY" was expressed by the continuous alphabets 'M', 'A', 'R' and 'Y'. The proposed method can adapt to significant motion variations among the signs, fingerspellings and non-sign patterns, and the BoostMap embeddings can adapt to significant shape variations, which enables ASL sign spotting even in a difficult case of utterance data. The frame-wise fingerspelling inference results are presented.

Table 8 shows the signs and fingerspellings spotting results for the signed utterances described in Table 3. The second and third columns represent the actual number of sign and fingerspelling segments in the input sequence, respectively. The fourth and fifth columns represent the respective ground-truth sign and fingerspelling segments in the input sequence as frame numbers. Each segment is represented with the set of frame numbers of the beginning and end points. There is little difference between the actual and spotted segments, because the transition movements appear near the boundaries of the signs and fingerspellings. As shown in the result, the proposed method can discriminate signs, fingerspellings and non-sign patterns correctly.

Fig. 14 presents a fingerspelling spotting result which has been classified by the letters 'I', 'O', 'H' and 'N'. As mentioned in Section 4.5, there is an ambiguity between the letters 'I' and 'J'. In order to eliminate this ambiguity, a conventional CRF is applied. The letter 'I' was replaced with 'J' by verifying the trajectory of the hand motion, as shown in the left graph of Fig. 14.

6. Conclusions and further research

In this paper, a new approach for simultaneous spotting of signs and fingerspellings was proposed. It used a hierarchical framework consisting of three steps: (1) Candidate segments of signs and fingerspellings were detected using a two-layer CRF, which consisted of a T-CRF and a conventional CRF. The T-CRF, which selected the label with maximum probability, can discriminate signs, fingerspellings and non-sign patterns using the combination of the hand motion and place of articulation, and the conventional CRF can recognize subsign patterns between signs. (2) Hand shapes of detected signs and fingerspellings were verified using BoostMap embeddings. (3) The motions of fingerspellings were verified in order to distinguish fingerspellings which had similar hand shapes and different hand trajectories.

Experiments demonstrated that the proposed method can spot signs and fingerspellings from utterance data at rates of 83% and 78%, respectively. This paper demonstrated that the proposed hierarchical framework with hierarchical CRF and BoostMap embeddings can accurately spot the combination of signs and fingerspellings.

Near-term future work will extend the proposed system so that it can recognize non-manual signs such as facial expressions.

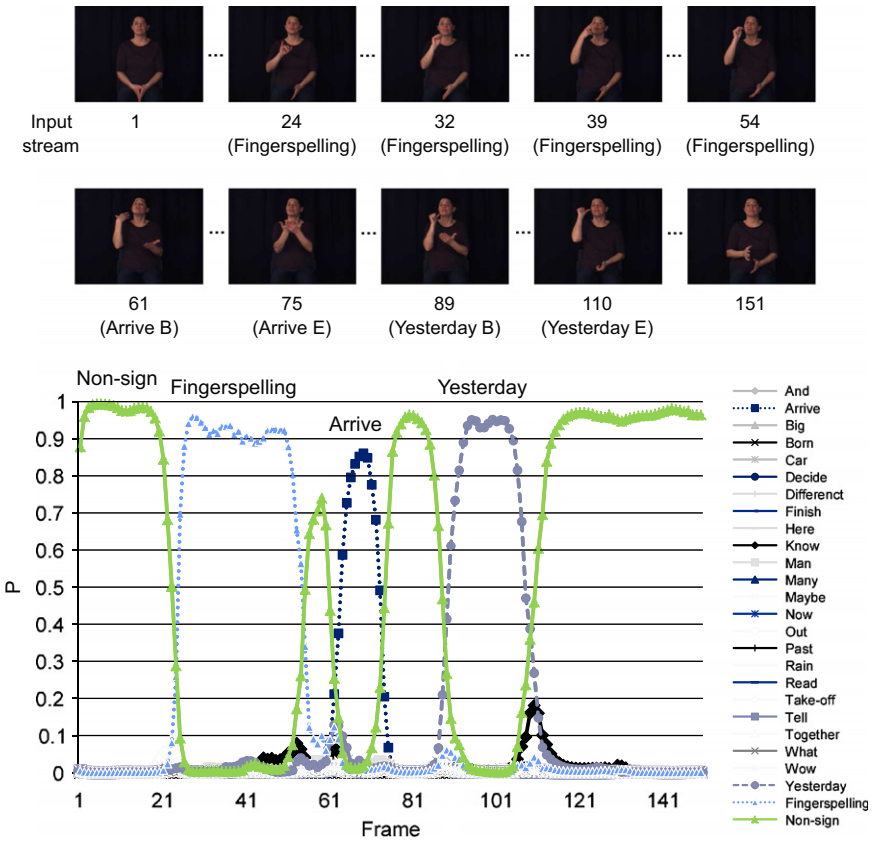

Fig. 12. Sign and fingerspelling spotting results for one ASL utterance sentence, "MARY arrived yesterday".

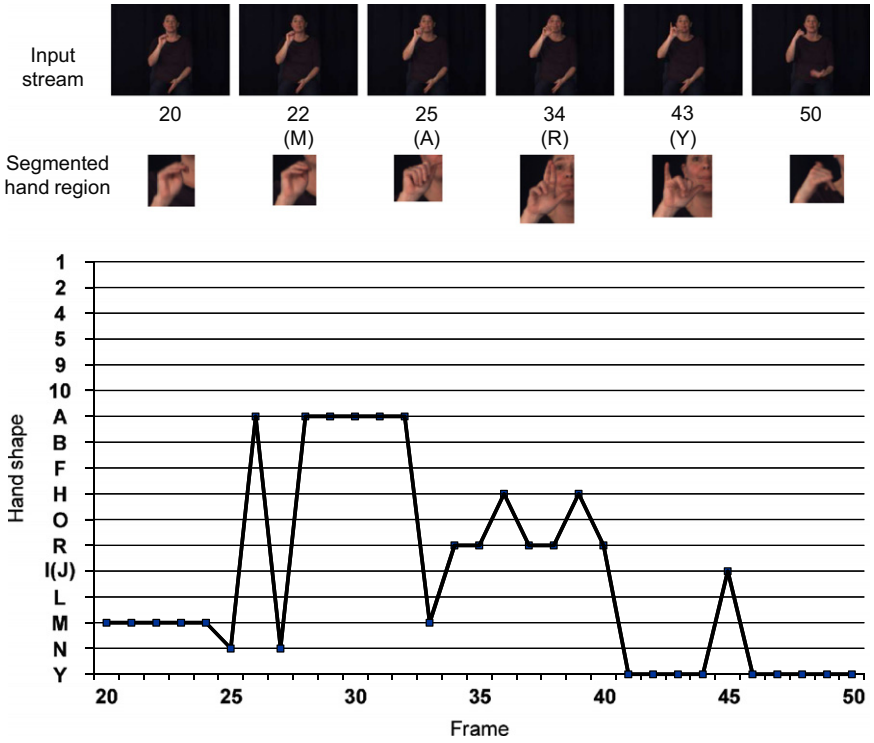

Fig. 13. An example demonstrated on a test sequence representing the word "MARY".

Table 8
Examples of sign and fingerspelling spotting results for three ASL utterance sentences.

Videos	Number of sign segments	Number of fingerspelling segments	Ground-truth for signs	Ground-truth for fingerspellings	Spotted sign segments	Spotted fingerspelling segments
1	2	1	(61–75) (89–110)	(24–54)	(65–74) (90–111)	(26–55)
2	3	1	(51–58) (76–87) (99–114)	(17–37)	(44–62) (84–90) (90–108)	(15–44)
3	4	1	(57–71) (83–101) (109–124) (136–160)	(24–46)	(50–76) (89–90) (104–121) (130–170)	(20–49)

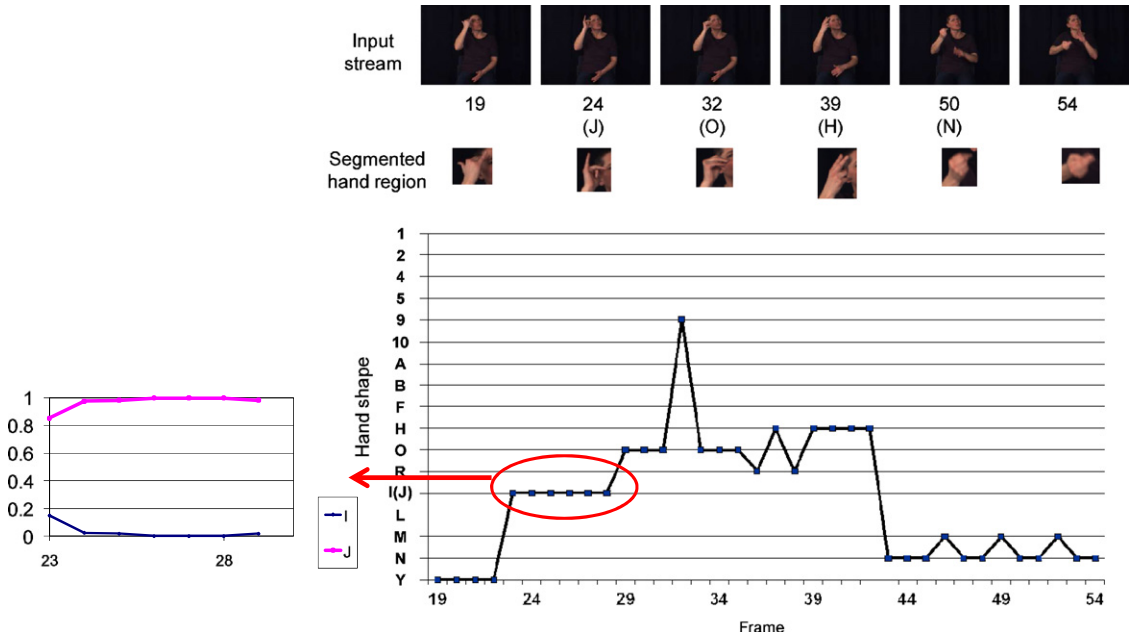

Fig. 14. A fingerspelling spotting example demonstrated on a test sequence representing the word "JOHN".

Acknowledgments

This research was supported by WCU (World Class University) Program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (R31-2008-000-10008-0). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (no. 2009-0086841) and the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy of Korea.

References

[1] S. Liwicki, M. Everingham, Automatic recognition of fingerspelled words in British sign language, in: Proceedings of the Workshop on CVPR for Human Communicative Behavior Analysis, Miami, USA, pp. 50–57.
[2] F. Lewis, Focus on Nonverbal Communication Research, Nova Science Publishers, 2007, pp. 79–100.
[3] B. Parton, Sign language recognition and translation: a multidisciplined approach from the field of artificial intelligence, Journal of Deaf Studies and Deaf Education 11 (2005) 94–101.
[4] L. Ding, A. Martinez, Modelling and recognition of the linguistic components in American sign language, Image and Vision Computing 27 (2009) 826–844.
[5] G. Tsechpenakis, D. Metaxas, C. Neidle, O. Hadjiladis, Robust online change point detection in video sequences, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshop, New York, USA.
[6] G. Tsechpenakis, D. Metaxas, C. Neidle, Learning-based dynamic coupling of discrete and continuous trackers, Computer Vision and Image Understanding 104 (2006) 140–156.
[7] T. Starner, J. Weaver, A. Pentland, Real-time American sign language recognition using desk and wearable computer based video, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 1371–1375.
[8] C. Vogler, D. Metaxas, A framework for recognizing the simultaneous aspects of American sign language, Computer Vision and Image Understanding 81 (2001) 358–384.
[9] R. Yang, S. Sarkar, Detecting coarticulation in sign language using conditional random fields, in: Proceedings of the International Conference on Pattern Recognition, Hong Kong, China, pp. 108–112.
[10] R. Yang, S. Sarkar, B. Loeding, Enhanced level building algorithm for the movement epenthesis problem in sign language recognition, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, Minnesota, USA, pp. 1–8.
[11] H.-D. Yang, S. Sclaroff, S.-W. Lee, Sign language spotting with a threshold model based on conditional random fields, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009) 1264–1277.

[12] H.-K. Lee, J.-H. Kim, An HMM-based threshold model approach for gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 961–973.
[13] R. Feris, M. Turk, R. Raskar, K.-H. Tan, G. Ohashi, Exploiting depth discontinuities for vision-based fingerspelling recognition, in: Proceedings of the Workshop on Real-Time Vision for Human–Computer Interaction, Washington, USA.
[14] P. Goh, E. Holden, Dynamic fingerspelling recognition using geometric and motion features, in: Proceedings of the International Conference on Image Processing, Atlanta, USA, pp. 2741–2744.
[15] V. Athitsos, J. Alon, S. Sclaroff, G. Kollios, Boostmap: an embedding method for efficient nearest neighbor retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 89–104.
[16] L. Gui, J. Thiran, N. Paragios, Finger-spelling recognition within a collaborative segmentation/behavior inference framework, in: Proceedings of European Signal Processing Conference, Lausanne, Switzerland, pp. 2741–2744.
[17] C. Ong, S. Ranganath, Automatic sign language analysis: a survey and the future beyond lexical meaning, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 873–891.
[18] G. Hjaltason, H. Samet, Properties of embedding methods for similarity searching in metric spaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 530–549.
[19] J. Lichtenauer, E. Hendriks, M. Reinders, Sign language recognition by combining statistical DTW and independent classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 2040–2046.
[20] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 509–522.
[21] E. Keogh, Exact indexing of dynamic time warping, in: Proceedings of the International Conference on Very Large Data Bases, Hong Kong, China, pp. 406–417.
[22] G. Mori, S. Belongie, J. Malik, Efficient shape matching using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1831–1837.
[23] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[24] H. Barrow, J. Tenenbaum, R. Bolles, H. Wolf, Parametric correspondence and chamfer matching: two new techniques for image matching, in: Proceedings of the International Joint Conference on Artificial Intelligence, Cambridge, USA, pp. 2741–2744.
[25] J. Bourgain, On Lipschitz embedding of finite metric spaces in hilbert space, Israel Journal of Mathematics 52 (1985) 46–52.
[26] D. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the International Conference on Computer Vision, Kerkyra, Greece, pp. 1150–1157.
[27] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the International Conference on Machine Learning, Williamstown, USA, pp. 282–289.
[28] H. Wallach, Conditional random fields: an introduction, Technical Report MS-CIS-04-21, University of Pennsylvania, 2004.
[29] R. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (1999) 297–336.
[30] V. Athitsos, Learning embeddings for indexing, retrieval and classification, with applications to object and shape recognition in image databases, Ph.D. Dissertation, Boston University, 2006.
[31] P. Dreuw, C. Neidle, V. Athitsos, S. Sclaroff, H. Ney, Benchmark databases for video-based automatic sign language recognition, in: Proceedings of the International Conference on Language Resources and Evaluation, Marrakech, Morocco, pp. 1115–1120.
[32] P. Viola, M.J. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2004) 137–154.
[33] J. Alon, V. Athitsos, Q. Yuan, S. Sclaroff, A unified framework for gesture recognition and spatiotemporal gesture segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2009) 1685–1699.
[34] H.-D. Yang, S.-W. Lee, S.-W. Lee, Multiple human detection and tracking based on weighted temporal texture features, International Journal of Pattern Recognition and Artificial Intelligence 20 (2006) 377–391.
[35] Poser 5 reference manual, Curious Labs, Inc., 2004.
[36] R. Battison, Lexical Borrowing in American Sign Language, Linstok Press, 1978, pp. 79–100.

About the Author—HEE-DEOK YANG received the B.S. degree in Computer Science from Chungnam National University, Daejeon, Korea, in 1998, and the M.S. and Ph.D. degrees in Computer Science and Engineering from Korea University, Seoul, Korea, in 2003 and 2008, respectively. He is currently an Assistant Professor of the School of Computer Engineering at Chosun University, Gwangju, Korea. His research interests include sign language recognition, gesture recognition, and brain engineering.

About the Author—SEONG-WHAN LEE received the B.S. degree in Computer Science and Statistics from Seoul National University, Seoul, Korea, in 1984, and the M.S. and Ph.D. degrees in Computer Science from KAIST in 1986 and 1989, respectively. From 1989 to 1995, he was an Assistant Professor in the Department of Computer Science at Chungbuk National University, Cheongju, Korea. In 1995, he joined the Faculty of the Department of Computer Science and Engineering at Korea University, Seoul, Korea, as an Associate Professor, and he is now a Full Professor. In 2009, he was appointed as the Hyundai-Kia Motor Chair Professor. He was the winner of the Annual Best Paper Award of the Korea Information Science Society in 1986. He obtained the First outstanding Young Researcher Award at the Second International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Award from Chungbuk National University in 1994. He also obtained the Outstanding Research Award from the Korea Information Science Society in 1996. He is a Fellow of IEEE, IAPR, and Korean Academy of Science and Technology. His research interests include pattern recognition, computer vision, and brain engineering. He has more than 250 publications in international journals and conference proceedings, and has authored ten books.