Pattern Recognition 47 (2014) 1294–1308

Contents lists available at ScienceDirect

Pattern Recognition

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

Towards subject independent continuous sign language recognition: A segment and merge approach

W.W. Kong [a], Surendra Ranganath [b],*

[a] Department of Electrical & Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576, Singapore
[b] Department of Information Science and Engineering, Sri Jayachamarajendra College of Engineering, Mysore 570002, India

ARTICLE INFO

Article history:
Received 2 June 2012
Received in revised form
12 September 2013
Accepted 21 September 2013
Available online 30 September 2013

Keywords:
Gesture recognition
Sign language recognition
Signer independence
Bayesian network
Conditional random field (CRF)
Support vector machine (SVM)
Semi-Markov CRF
Hidden Markov model (HMM)

ABSTRACT

This paper presents a segment-based probabilistic approach to robustly recognize continuous sign language sentences. The recognition strategy is based on a two-layer conditional random field (CRF) model, where the lower layer processes the component channels and provides outputs to the upper layer for sign recognition. The continuously signed sentences are first segmented, and the sub-segments are labeled *SIGN* or *ME* (movement epenthesis) by a Bayesian network (BN) which fuses the outputs of independent CRF and support vector machine (SVM) classifiers. The sub-segments labeled as *ME* are discarded and the remaining *SIGN* sub-segments are merged and recognized by the two-layer CRF classifier; for this we have proposed a new algorithm based on the semi-Markov CRF decoding scheme. With eight signers, we obtained a recall rate of 95.7% and a precision of 96.6% for unseen samples from seen signers, and a recall rate of 86.6% and a precision of 89.9% for unseen signers.

1. Introduction

In recent years, there has been increasing interest in developing automatic sign language recognition systems to facilitate communication between the deaf and hearing people. In manual signing, four components are used to compose signs, namely, handshape, movement, palm orientation and location; the systematic change of these components produces a large number of different sign appearances. The appearance and meaning of basic signs are well-defined in sign language dictionaries; however, in practice, variations arise due to regional, social, and ethnic factors, and also from gender, age, education and family background. This can lead to significant variations in manual signs performed by different signers, and pose challenging problems for developing robust computer-based sign language recognition systems.

1.1. Variations and movement epenthesis in manual signing

Variations which appear in the basic components, i.e. handshape, movement, palm orientation and location, are classified as phonological variation by linguists. Some handshapes are naturally close to each other; for example, the signs with handshapes “S” and “A” in American sign language (ASL) can look very similar if they are signed loosely. Also, some handshapes may be used interchangeably in certain signs, for example, signs such as FUNNY, NOSE, RED and CUTE are sometimes signed with or without thumb extension [1]. Studies in [2] show that ASL signs with handshape “1” (index finger extended, all other fingers and thumb closed) are very often signed as signs with handshape “L” (thumb and index extended, all other finger closed) or handshape “5” (all fingers open).

Locations of signs may also change. For example, the ASL sign for KNOW is prescribed to be signed at the forehead, but it is frequently signed at a lower position near the cheek. In [2], it was found that younger signers tend to make these signs below the forehead more often than older signers. Also, men tend to lower the sign location more than women. The movement path and palm orientation of a sign may also be modified; for example, signs with straight line movement can often be signed with arc-shaped movement or with palm orientation changing from palm-down to palm-left. Assimilation of handshape, movement, palm orientation and location also occur in compound signs. This refers to a process when two signs forming a compound sign begin to look similar. Some other phonological variations include deletion of one hand in a two-handed sign and hand contact.

There can be systematic variations present in the grammatical aspect of sign language. Typological variations concerning sign order

* Corresponding author. Tel: +91 900 8400 711.
E-mail addresses: wwn.kong@gmail.com (W.W. Kong), surendra@iitgn.ac.in (S. Ranganath).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.09.014

also occur where signs are arranged differently in sentences. Lastly, some signs can be made in unpredictable ways. For example, in [2] it was reported that the sign for PIZZA was fingerspelled by some, others signed it iconically as a person taking a bite out of a pizza, or as a round plate on which pizza is served. These variants of the sign do not share handshapes, locations, orientations and movements.

The above variations are related to linguistic aspects, and a sign language recognition system involving multiple signers must robustly handle these variations. In addition, physical variations (e.g. hand size, body size, length of the arm, etc. of the signers) also contribute to the complexity of building a robust recognition system.

Movement epenthesis (ME), the transition segment which connects successive signs, is formed when the hands move from the ending location of one sign to the starting location of the next sign, and does not carry any sign information. Linguistic studies of ME in the literature are limited and it does not have a well-defined lexicon. Perlmutter [3] also showed that ME had no phonological representation. As a connecting segment between signs, its starting and ending locations would depend on the preceding and succeeding signs, and its duration could even be comparable to that of a sign segment. Also, variations in adjacent signs may affect the ME, and it is possible that the variations in ME are comparable to variations in sign. As there are no well-defined rules for making such transitional movements, dealing with ME adds significant complexity to the task of recognizing continuous signs. This problem needs to be addressed explicitly for robust sign language recognition. It must be noted that ME is a different phenomenon from co-articulation in speech; co-articulation does occur in sign language, and manifests itself in some signs as hold deletion, metathesis and assimilation [2].

1.2. Motivation and scope

For sign language communication to be natural and effective, deviations from textbook definitions of sign can be expected. Hence, a practical sign language recognition system must be robust to these variations across signers. In the literature, most works which deal with signer independence issues consider hand postures or isolated signs [4] but works on signer independence with continuously signed sentences are limited. Some alternative approaches rely on an adaptation strategy, where a trained system is adapted to a new signer by collecting a small amount of signer specific data. Though adaptation is a reasonable approach, a truly signer independent system would be ideal.

Many recent works have considered recognition of continuously signed sentences, but their main focus has been on obtaining high recognition accuracy and scalability to large vocabulary. These are important problems to consider; however, several of these works report results based on only one signer. We consider recognizing continuously signed sentences from several signers, with resulting increase in inter-signer variations. In this paper we consider the phonological variations in sign language, i.e. variations in handshape, movement, palm orientation and location, arising from natural signing. We also include directional verbs which exhibit variation in grammatical aspect, and variations in sign order which can occur in natural signing.

In addition, inter-signer variations in ME also pose a challenge for accurate sign recognition. However, many works either neglect it or pay no special attention to the problem. In works that do consider it explicitly, the common approaches are either to model ME explicitly, or assume that the ME segments can be absorbed into adjacent signs. In this paper, we account for ME explicitly, though without elaborate modeling of these “extraneous” segments.

In the following, Section 2 summarizes related works. The overview of our proposed strategy for handling signer variation and ME is described in Section 3. Section 4 presents the feature representation used in the recognition framework. In Section 5, we discuss the strategy to deal with ME and present a conditional random field/support vector machine (CRF/SVM) and Bayesian network (BN) based classifier to discriminate between sign and ME. Sections 6 and 7 describe the complete recognition framework based on a two-layer CRF model and its decoding algorithm, respectively. Experimental results are presented in Section 8, and include comparisons with hidden Markov models (HMMs) along with results, analysis and discussion. Lastly, Section 9 gives the conclusions of this paper and suggestions for future work.

2. Related works

Recently, some works have considered the ME problem. Yang and Sarkar [5,6] adopted an enhanced level building algorithm (eLB) which was used to simultaneously segment and match signs to continuous sign language sentences. ME was automatically discarded during the matching process. They enhanced the classical level building algorithm [7] based on dynamic programming, and coupled it with a trigram grammar, to obtain 83.0% sign level recognition in [5]. Further experiments in [6] on ASL data sets showed that their approach outperformed CRFs and latent dynamic CRF-based approaches. In the works by Lee et al. [8–12], sign spotting in continuously signed sentences was used to recognize signs. They first trained a CRF model with only sign samples. Then, a threshold model with CRF was derived by adding the label for non-sign patterns by using the weights of the state and transition feature functions of the original CRF. ME segments were bypassed automatically. Testing on continuous ASL sentences consisting of 48 signs yielded 87.0% spotting rate and 93.5% recognition rate on the spotted isolated signs. Later extensions to spot signs and fingerspellings simultaneously using hierarchical CRFs [11,12] yielded 83.0% and 78.0% spotting rate for signs and fingerspellings, respectively. Kelly et al. [13,14] also proposed a parallel HMM threshold model to handle ME based on the threshold HMM proposed by Lee and Kim [15]. The key idea in threshold HMM is to use the likelihood as an adaptive threshold for selecting the proper gesture model. Kelly et al. [14] reported that 100 different types of ME and eight different signs were identified in experiments.

Works such as [5,6,10,14] used only signs for training and dealt with ME during the decoding process to avoid modeling the latter explicitly. However, Yang et al. [5,6] only used a single channel for processing and recognition, making the scheme vulnerable to signer variations, and limiting generalization to new signers. In their experiments with three signers, recognition results for a new signer were inconsistent. They reported accuracy of 80%, slightly more than 50% and less than 30% for three rounds of leave-one-out experiments with 10 sentences. The limited generalization could also be due to the generative modeling used for signs. Furthermore, their sign based modeling approach may not be scalable to large vocabulary compared to a phoneme-based approach. Both of the other works [10,14] were based on threshold models trained with only one signer, with parameters that were derived from the training data. However, finding good threshold values may be difficult when the problem is extended to several signers and the recognition framework may not perform robustly with new signers.

A practical continuous sign language recognition system needs to be signer independent; however, this has not received much attention in the literature. Presently, several continuous sign language recognition works, e.g. [10,16–21] and some of the works mentioned above mainly report results on a single signer. Two strategies for signer independent recognition are to (1) build a baseline recognition system with a few signers and adapt the system to a new signer using a small amount of training data, or (2) devise a robust recognition algorithm that is designed to be tolerant to signer variations and thus yield good generalization.

Fang [22,23] demonstrated some signer independent attributes in their continuous Chinese sign language (CSL) recognition systems using a hybrid of recurrent neural network and HMMs. Three signers signed 100 sentences consisting of 208 signs twice. They used partial data from two signers for training and left out one signer as "unseen" by the system. They showed recognition accuracy of 85.0% for the unseen signer while the standard HMM approach showed 81.2%. However, the nature of the signs and sentences used in their works is not clear, nor how their methodology adequately addresses signer independence. Farhadi et al. [24] proposed a somewhat different approach based on transfer learning which relied heavily on the discriminative features describing the intrinsic properties of a sign. Logistic regression was used to spot the word boundaries, and discriminative feature spaces based on the dictionary were used to compare with the test image feature spaces. They trained their system to recognize 90 signs and tested it on a new signer signing 40 signs in frontal view as well as 3/4 view, obtaining 64.2% and 62.5%, accuracy, respectively.

In works that use an adaptation strategy to generalize to a new signer, Ong and Ranganath [25] adapted trained BNs by a maximum a posteriori (MAP) technique to recognize 20 simulated isolated sign gestures. Data from three signers was used to train the BNs, and one new signer was used for testing. Accuracies of 52.6% and 88.5% were obtained for experiments without adaptation and with adaptation, respectively. von Agris et al. [26] devised a vision-based recognition system for 153 isolated signs that adapted to unseen signers, based on maximum likelihood linear regression (MLLR) and MAP estimation. Three signers were used for training the signer independent model, and one signer was used for testing. Supervised adaptation with 80 adaptation sequences yielded an accuracy of 78.6% while the signer independent system without adaptation yielded 55.5%. They subsequently considered continuous signing in [27,28]. In [27], a database of sentences in German sign language was created, comprising 450 basic signs making up 780 sentences from 20 different signers. Preliminary recognition experiments were conducted based on the HMM framework. For signer independent experiments with adaptation, accuracies of 70.4%, 67.8%, and 64.9% were reported for vocabulary sizes of 150, 300 and 450, respectively. In a more comprehensive work [28], they combined eigenvoice (EV), MLLR, and MAP to adapt trained HMMs. MLLR and MAP were the basic adaptation strategies, and the eigenvoice approach [29] provided constraints to reduce the number of free parameters to be adapted. The EV+MLLR+MAP adaptation provided the best results yielding 75.8% accuracy, while baseline HMMs yielded 65.3% in leaving-one-out tests on 25 native signers.

3. Overview

We propose a two-layer multichannel system for recognizing continuously signed ASL sentences. In our work, we used Cyberglove and magnetic trackers for data acquisition. The overall block diagram of the system is shown in Fig. 1, and described below. We first segment the continuous input sequences using a segmentation algorithm from our previous work [30] based on minimum velocity and maximum directional angle change in the movement channel. All the (four) channels are assumed to evolve synchronously, and hence, the same segment boundary points are assigned to the other three channels also. This yields the component sequences, $\{h_i, m_i, o_i, l_i\}$, $i = 1, \ldots, n$ with n sub-segments. The detection rate for the true sign boundary points is high, about 99.9%, but this is also accompanied by a high false alarm rate, i.e. over-segmented sequences are produced. However, with this high detection rate, we anticipate that if the sub-segments can be correctly labeled as belonging to sign (*SIGN*) or ME (*ME*) and the *SIGN* sub-segments are merged properly while discarding the *ME* sub-segments, then the sign sequence in a sentence can be decoded with high accuracy. Thus, we adopt a segment-based approach to recognize the continuously signed sentences, rather than the usual frame-based approach.

Following the segmentation module is a high accuracy classifier that assigns a label $\ell_i = \{SIGN, ME\}$ to each sub-segment in the movement channel. Due to the assumption of synchronous channels, the same labels are also applied to the corresponding sub-segments of the other three channels. These labeled sub-segment sequences in the four channels are output from the classifier module as shown in Fig. 1 (e.g. ${}^{\ell_i}m_i$, etc.). The *SIGN*/*ME* labeling is done by a BN classifier that fuses the results of independent CRF and SVM classifiers. Following this, the *SIGN* sub-segments (e.g. ${}^{SIGN}m_i$) are retained for sign decoding; the *ME* sub-segments are discarded, though their locations are recorded, as they provide useful information to the final decoding algorithm for reducing computations.

We then work out a strategy to merge the *SIGN* sub-segments and recognize the signs with a two-layer CRF model as shown in Fig. 1. The lower layer of the recognition module consists of four independent linear CRF models to recognize and output sequences of phoneme labels (e.g. y_{m_i}) in the component channels, from the corresponding *SIGN* sub-segment sequences. The corresponding phoneme labels from the four channels are then combined and input to the upper layer semi-Markov CRF for sign recognition. For inferring the sign label, we modified the decoding algorithm of the semi-Markov CRFs proposed by Sarawagi and Cohen [31] for our two-layer multichannel scheme. The modification also enabled the inclusion of two-class SVMs to identify incorrectly merged sub-segments and to accommodate skip states for dealing with incorrectly labeled *ME* sub-segments. During decoding, multiple, contiguous of sub-segments are merged into segments, features are extracted from them, and the best merged segments and their associated sign labels are computed using an efficient Viterbi-like algorithm. The two-layer recognition system is trained using only sign segments after removing all the ME segments manually.

4. Feature normalization and representation

The raw data obtained from the glove and trackers (described in Section 8) consists of handshape (16-D vectors), palm orientation (9-D vectors) and position (3-D vectors), obtained at frame rates. These raw vectors need to be appropriately normalized prior to feature extraction. The 16-D handshape vectors are normalized to unit length to discount hand size variations. The raw position vectors yield data for the movement and location channels. A few vectors at the beginning and end of a segment are used to obtain location channel information.

In the movement channel, we have used movement direction and trajectory shape as the basic descriptors, rather than raw position vectors. These descriptors need to be invariant to location and size of trajectory, and hence care is required when obtaining them from the raw position data. For example, the position vectors for a circular hand movement made in the chest area will be different for the same movement made in the head area, and hence, normalization is necessary. However, in continuous signing, direct normalization based on the entire signed sentence can lead to errors as there may be variations from one sign to the next. On the other hand, normalization based on segments requires the segment start and end points which are unknown. We address

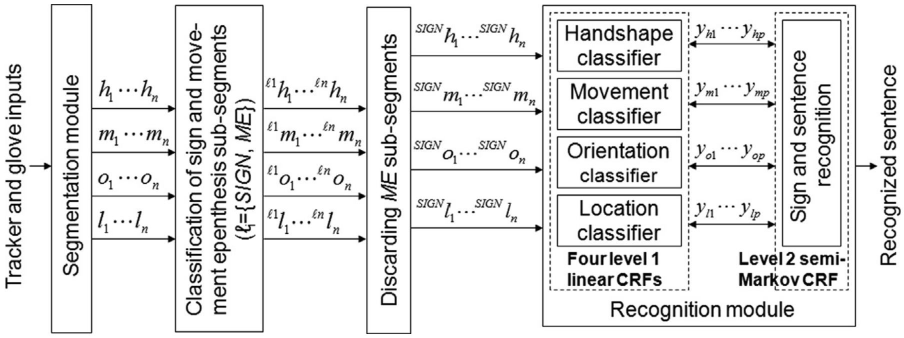
Tracker and glove inputs
Segmentation module
$h_1 \cdots h_n$
$m_1 \cdots m_n$
$o_1 \cdots o_n$
$l_1 \cdots l_n$
Classification of sign and movement epenthesis sub-segments (ℓ={SIGN, ME})
$^{\ell 1}h_1 \cdots {}^{\ell n}h_n$
$^{\ell 1}m_1 \cdots {}^{\ell n}m_n$
$^{\ell 1}o_1 \cdots {}^{\ell n}o_n$
$^{\ell 1}l_1 \cdots {}^{\ell n}l_n$
Discarding ME sub-segments
$^{SIGN}h_1 \cdots {}^{SIGN}h_n$
$^{SIGN}m_1 \cdots {}^{SIGN}m_n$
$^{SIGN}o_1 \cdots {}^{SIGN}o_n$
$^{SIGN}l_1 \cdots {}^{SIGN}l_n$
Handshape classifier
Movement classifier
Orientation classifier
Location classifier
Four level 1 linear CRFs
$y_{h1} \cdots y_{hp}$
$y_{m1} \cdots y_{mp}$
$y_{o1} \cdots y_{op}$
$y_{l1} \cdots y_{lp}$
Sign and sentence recognition
Level 2 semi-Markov CRF
Recognition module
Recognized sentence

Fig. 1. Proposed ASL recognition system.

these problems by fitting lines to the trajectory of each sub-segment using the iterative end-point fitting algorithm [32], and representing the trajectory by the unit directional vectors of the lines computed at every sample point (the collection of these vectors over all sub-segments represents the entire sentence trajectory).

These preprocessing steps yield a sequence of frame-based vectors in the four channels where the handshape vector has been normalized for hand size, and the movement component is represented by a sequence of directional unit vectors at every frame instant. The orientation vectors, and the location channel (represented by position vectors) are unchanged. However, for convenience, we refer to all these as the set of "normalized" vectors.

5. Subsystem I: Labeling of sign and movement epenthesis sub-segments

We first segment the continuous input sequences using a segmentation algorithm from our previous work [30] based on minimum velocity and maximum directional angle change in the movement channel. The goal here is to label these sub-segments as *SIGN* or *ME*. The classifier we propose is obtained by fusing the outputs of independent SVM and CRF classifiers by a BN.

5.1. Feature extraction

The features extracted from the sub-segments in the four parallel channels are listed and described in Table A1 (Appendix). The state and transition features used for the CRF, and the features used for the SVM are given in the CRF and SVM1 columns of Table A1. Some other details are described in the following. The "start" and "end" features of each component are computed by taking the mean of the first and last 5% of data, respectively, from the sub-segments. The dominant direction of a sub-segment in the hand movement channel is obtained as the principal eigenvector of the sub-segment as described in our previous work [33]. For the CRF, trigram features refer to triplets that consist of the feature from the current sub-segment and from the previous (or future) two sub-segments. For example, for a sequence of handshape sub-segments, $\mathbf{h} = (h_1, h_2, h_3, \ldots, h_T)$, the **tri_feature** formed at sub-segment 3 are "$h_1h_2h_3$" and "$h_3h_4h_5$". Tri features can be formed from all CRF state features except for arc length. The other features listed in Table A1 for the CRF and SVM are self-explanatory.

The number of discrete symbols for each CRF feature was obtained experimentally by clustering the data vectors of that feature (e.g. **hand_start** which is 16-D) into $\hat{k}$ clusters using k-means. To determine the best value for $\hat{k}$, we trained a CRF for different k values using only the target feature, e.g. **hand_start**, and picked the $\hat{k}$ that yielded the highest CRF classification accuracy (the values obtained are shown in the *#Sym* column of Table A1). For the SVM, all the individual component features were cascaded to form a 126-D feature vector for input to the classifier.

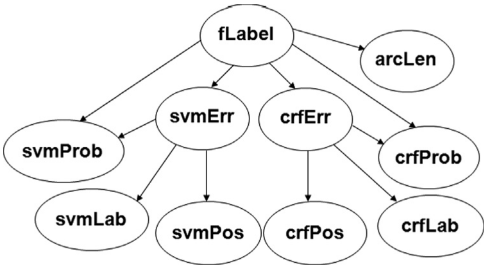

Fig. 2. BN classifier to label sub-segments as *SIGN*/*ME* by fusing CRF and SVM outputs.

5.2. Sub-segment classification

To provide a good starting point for the decoding algorithm, the sub-segments should be labeled as *SIGN*/*ME* with the highest possible accuracy. With this motivation, we compared a SVM with probabilistic output [34] and a standard linear CRF [35,36]. The CRF and SVM both used essentially the same set of basic features extracted from the four component sub-segments (see Table A1); however, the features for the SVM were continuous-valued features while the features for the CRF were discretized into a finite symbol set. The settings and procedures for training the CRF and SVM are described in Section 8.

Experiments showed that while both classifiers provided good accuracy, there was scope for improvement. To achieve this, we computed the SVM and CRF output probabilities and fused them (along with other useful features) using a BN. We defined three query nodes for the BN viz. **fLabel**, **svmErr**, and **crfErr**, and seven observed nodes viz. **svmProb**, **svmLab**, **svmPos**, **crfProb**, **crfLab**, **crfPos** and **arcLen**. The structure of the BN is shown in Fig. 2 and was specified using prior knowledge; all the nodes have finite discrete states which are described in Table A2 (Appendix). The observed nodes **svmPos** and **crfPos** are defined to have four states, which are described through the following example. Given an SVM or CRF labeled sequence of sub-segments, $\mathbf{s}=\{^{SIGN}s_1, {}^{ME}s_2, {}^{ME}s_3, {}^{SIGN}s_4, {}^{SIGN}s_5, {}^{SIGN}\ s_6\}$, four categories for the positions of the sub-segments within the sequence can be identified by grouping together sub-segments having the same consecutive labels: $\{^{SIGN}s_1\}$, $\{^{ME}s_2, {}^{ME}s_3\}$, $\{^{SIGN}s_4, {}^{SIGN}s_5, {}^{SIGN}s_6\}$. The first group with only one sub-segment $^{SIGN}s_1$ is labeled as "single position". To distinguish sub-segments lying at the group edges, $^{ME}s_2$ and $^{SIGN}s_4$ are labeled as "left position" while $^{ME}s_3$ and $^{SIGN}s_6$ are labeled as "right position". The sub-segment, $^{SIGN}s_5$, which lies between edges is labeled as "other position". The aim is to use the error pattern based on the positions of the sub-segments within a group to improve accuracy.

The parameters of the BN in Fig. 2 (conditional probability table (CPT) of the nodes) are obtained by ML estimation from given training data. In the final sign decoding step, loss of sign sub-segments through misclassification is more problematic for sign recognition compared to false alarms (*ME* sub-segments being classified as *SIGN*). Hence, we adjusted the classification threshold to minimize the number of missed *SIGN*s at the expense of having more false alarms. Finally, all sub-segments labeled as *ME* are discarded, though their positions are recorded for use in sign recognition.

6. Subsystem II: Segmental sign recognition

As outlined in Section 3, sign recognition is implemented by a two-layer CRF model. The lower layer of parallel CRFs uses the standard decoding algorithm; however, for the upper layer CRF which fuses the phoneme label outputs of the lower layer CRFs to recognize signs, we modified the decoding algorithm of the semi-Markov CRFs proposed by Sarawagi and Cohen [31], who were concerned with named entity recognition in text sentences. There, a sequence of words (analogous to sub-segments in our context) in a sentence was classified by a semi-Markov CRF into segments that consisted of a group of contiguous words representing named entities. For completeness, we briefly describe the model of [31] below, and motivate the modifications necessary to devise an efficient decoding algorithm for the continuous sign recognition problem.

For each sentence, the input to the semi-Markov CRF model of [31] is a sequence, $\mathbf{s}=\{s_1,\dots,s_n\}$ consisting of n words (sub-segments in our context). Feature vectors are extracted from the words to form the corresponding input observation sequence $\mathbf{x}=\{x_1,\dots,x_n\}$, where x_j refers to a feature vector extracted from a word. Now, let $\mathbf{S}=\{S_1,S_2,\dots,S_p\}$ denote a sequence of segments (a segment refers to a set of contiguous elements of either $\mathbf{s}$ or $\mathbf{x}$) of $\mathbf{x}$, where $S_t=\{x_{u_t}:x_{v_t},y_t\}$ has start position u_t, end position v_t, with $1\le u_t\le v_t\le|\mathbf{x}|$ and a label $y_t\in\mathbf{Y}$. The length of the output label sequence $\mathbf{y}$ depends on the final number of segments obtained by merging words in $\mathbf{x}$. Defining a segment feature function as $g_i(\mathbf{S},\mathbf{x},t)=g_i(y_{t-1},y_t,\mathbf{x},u_t,v_t)$, $i=1,\dots,h$, the likelihood function is given as

$$p(\mathbf{S}|\mathbf{x})=\frac{1}{\mathbf{Z}(\mathbf{x})}\exp\left(\sum_{t=1}^{p}\sum_{i=1}^{h}\gamma_i g_i(\mathbf{S},\mathbf{x},t)\right),\tag{1}$$

where $\boldsymbol{\theta}=\{\gamma_1,\gamma_2,\dots,\gamma_h\}$ are parameters to be estimated and the normalizing factor $\mathbf{Z}(\mathbf{x})$ is given as

$$\mathbf{Z}(\mathbf{x})=\sum_{\mathbf{S}\in\mathcal{S}}\exp\left(\sum_{t=1}^{p}\sum_{i=1}^{h}\gamma_i g_i(\mathbf{S},\mathbf{x},t)\right).\tag{2}$$

Given training data, $\mathcal{D}=\{\mathbf{x}^k,\mathbf{S}^k\}_{k=1}^{\mathcal{K}}$, the log-likelihood with L_2-norm regularization is written as

$$\mathcal{L}(\boldsymbol{\theta})=\sum_{k=1}^{\mathcal{K}}\sum_{t=1}^{p}\sum_{i=1}^{h}\gamma_i g_i(\mathbf{S}^k,\mathbf{x}^k,t)-\sum_{k=1}^{\mathcal{K}}\log\,\mathbf{Z}(\mathbf{x}^k)-C\sum_{i=1}^{h}\frac{|\gamma_i|^2}{2}.\tag{3}$$

The parameter estimation procedure for semi-Markov CRFs is similar to conventional linear CRFs. The main difference is that the start and end positions of the segments are taken into consideration during training in semi-Markov CRFs (details can be found in [31]).

Inferencing in semi-Markov CRFs is formulated to find the best cost segment path given $\boldsymbol{\theta}$ and $\mathbf{x}$ as

$$\mathbf{S}^*=\underset{\mathbf{S}}{\operatorname{argmax}}\;p(\mathbf{S}|\mathbf{x})=\underset{\mathbf{S}}{\operatorname{argmax}}\;\exp\left(\sum_{t=1}^{p}\sum_{i=1}^{h}\gamma_i g_i(y_{t-1},y_t,\mathbf{x},u_t,v_t)\right).\tag{4}$$

Denoting the upper bound on segment length as L, ${}^{q}\mathbf{S}_{1:r}$ as the set of all possible segments in $\mathbf{x}'=\{x_1,\dots,x_r\}$, $r\le n$, such that the last segment has label q and ending position r, and $\eta_r(q)$ as the largest value of $p(\mathbf{S}'|\mathbf{x})$ for any $\mathbf{S}'\in{}^{q}\mathbf{S}_{1:r}$, the recursive formulation for estimating the best segment path in semi-Markov CRFs is similar to the Viterbi algorithm and is written as

$$\eta_r(q)=\begin{cases}\max\limits_{\substack{\hat{q}\\ d=1,\dots,L}}\eta_{r-d}(\hat{q})\Phi'_{(r-d+1):r}(\hat{q},q,\mathbf{x}) & \text{if } r>0,\\ 1 & \text{if } r=0,\\ 0 & \text{if } r<0,\end{cases}\tag{5}$$

where

$$\Phi'_{(r-d+1):r}(\hat{q},q,\mathbf{x})=\exp\left(\sum_{i=1}^{h}\gamma_i g_i(y_{t-1}=\hat{q},y_t=q,\mathbf{x},r-d+1,r)\right).\tag{6}$$

The best segment path is traced by backtracking from $\max_{q\in\mathbf{Y}}\eta_{|\mathbf{x}|}(q)$, along the maximum cost path.

In the named entity recognition problem, modeled by the semi-Markov CRFs proposed in [31], $\mathbf{x}$ is a sequence of feature vectors representing words. In that problem, high level features extracted from each word are used to segment the sequence. For example, capitalization of the first letter in consecutive words was used to indicate a segment of named entities. Such features can be extracted unambiguously and consistently from each word. However, direct use of semi-Markov CRF models for our problem would require each sub-segment to be treated analogous to a word. However, it is difficult to define sub-segment features that are similar to letter capitalization that can be consistently found to merge sub-segments into a segment. This is because in our sign recognition problem, firstly, extracted features are highly dependent on the initial segmentation of the sentence into sub-segments. In practice, we cannot expect identical break points or exactly the same number of sub-segments to occur in two samples of the same sentence (consisting of the same underlying sign segments which need to be recovered). Hence, if features are extracted directly from these sub-segment sequences, we may have very different representations of the observation sequences that are input to the semi-Markov CRF. Secondly, combining the features extracted from individual sub-segments may not be representative enough to characterize the signs. Thirdly, and more importantly, using the semi-Markov CRF for phoneme-based modeling independently in the four parallel component channels may lead to different segment lengths in the different channels after decoding, entailing a complex sign sequence decoding procedure.

Based on these observations, we propose the two layered CRF model for continuous sign recognition, where the lower layer consists of four linear CRFs to model phonemes in the component channels, and the upper layer is a semi-Markov CRF for sign recognition. The model can be trained using complete sign segments extracted from the signed sentences and using the conventional linear CRF learning algorithm as described in [35]. However, a new decoding algorithm for the two-layer CRF is needed to handle sub-segment merging and sign recognition. Our new algorithm modifies the decoding procedure of semi-Markov CRFs, to have the ability to merge sub-segments optimally for sign recognition. This requires merging the original sub-segments together and recomputing features from the merged sub-segments.

6.1. Training the two-layered CRF framework

Fig. 3 shows the two-layer sign recognition model, and the data hierarchy. $\tilde{H}_j$, $\tilde{M}_j$, $\tilde{O}_j$, and $\tilde{L}_j$ denote the jth segment in the

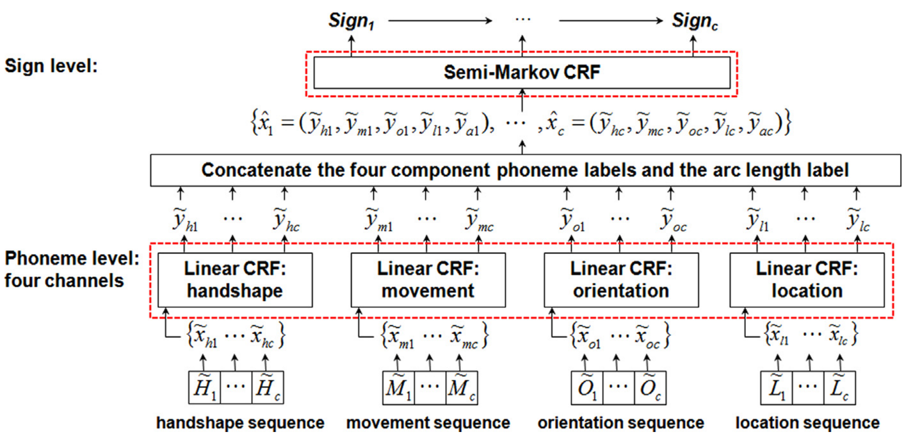
Sign level:
$Sign_1$
⋯
$Sign_c$
Semi-Markov CRF
$\{\hat{x}_1 = (\tilde{y}_{h1}, \tilde{y}_{m1}, \tilde{y}_{o1}, \tilde{y}_{l1}, \tilde{y}_{a1}), \cdots, \hat{x}_c = (\tilde{y}_{hc}, \tilde{y}_{mc}, \tilde{y}_{oc}, \tilde{y}_{lc}, \tilde{y}_{ac})\}$
Concatenate the four component phoneme labels and the arc length label
$\tilde{y}_{h1} \cdots \tilde{y}_{hc}$
$\tilde{y}_{m1} \cdots \tilde{y}_{mc}$
$\tilde{y}_{o1} \cdots \tilde{y}_{oc}$
$\tilde{y}_{l1} \cdots \tilde{y}_{lc}$
Phoneme level: four channels
Linear CRF: handshape
Linear CRF: movement
Linear CRF: orientation
Linear CRF: location
$\{\tilde{x}_{h1} \cdots \tilde{x}_{hc}\}$
$\{\tilde{x}_{m1} \cdots \tilde{x}_{mc}\}$
$\{\tilde{x}_{o1} \cdots \tilde{x}_{oc}\}$
$\{\tilde{x}_{l1} \cdots \tilde{x}_{lc}\}$
$\tilde{H}_1 \cdots \tilde{H}_c$
$\tilde{M}_1 \cdots \tilde{M}_c$
$\tilde{O}_1 \cdots \tilde{O}_c$
$\tilde{L}_1 \cdots \tilde{L}_c$
handshape sequence
movement sequence
orientation sequence
location sequence

Fig. 3. The two-layer CRF model for continuous sign recognition. CRFs in the four parallel channels of the lower layer use subphones to recognize corresponding component phonemes. These are fused by the semi-Markov CRF in the upper layer for sign recognition.

handshape, movement, orientation, and location channels, respectively, and are represented by the normalized vectors described in Section 4. These segments are processed to extract vectors of subphone labels, $\tilde{x}_{hj}$, $\tilde{x}_{mj}$, $\tilde{x}_{oj}$, and $\tilde{x}_{lj}$, and input to their respective linear CRF. Each CRF outputs a (scalar) phoneme label, $\tilde{y}_{hj}$, $\tilde{y}_{m_j}$, $\tilde{y}_{oj}$, and $\tilde{y}_{lj}$. These phoneme labels are concatenated into a vector, $\hat{x}_j$, along with a segment arc length label, $\tilde{y}_{aj}$, obtained from the movement channel, and input to the semi-Markov CRF in the upper layer to decode the sign for the jth segment, $sign_j$.

In the context of model training, the signed sentences from the training set need to be processed to extract appropriate data sets. We used a semi-automatic process to identify the sign and ME segments from the movement channel, and retained only the sign segments. The segment boundary points are assigned to the other channels as well, to maintain time alignment. Thus, the training data set consists of sequences of sign segments in sentences, and their associated sign labels. (We note here parenthetically, that in contrast to this ideal situation where sign segments are available, the recognition phase is complex as segments corresponding to signs are not known.)

During training, sign segments such as $\tilde{S}_j = \{\tilde{H}_j, \tilde{M}_j, \tilde{O}_j, \tilde{L}_j\}$, and their corresponding sign label ℓ_{sj}, are available. However, the labels, $\{\tilde{y}_{hj}, \tilde{y}_{mj}, \tilde{y}_{oj}, \tilde{y}_{lj}\}$, that need to be assigned to the corresponding component segments are not available a priori, and are discovered through a separate data driven clustering process. These labels can be considered as phonemes representing the sign, ℓ_{sj}. In turn, phonemes are represented by subphones; to discover subphone labels, sign segments from the training set are divided into M equal intervals, and the corresponding data is clustered. These details are discussed below.

6.1.1. *Training at the phoneme level*

The training sequences at the phoneme level are denoted as $\mathcal{D}_t = \{\tilde{\mathbf{x}}_t^k, \tilde{\mathbf{y}}_t^k\}$, with $t = h, m, o, l$, representing handshape, movement, orientation and location, for $k = 1, \ldots, N$, where $\tilde{\mathbf{x}}_t^k = \{\tilde{x}_{t1}^k, \ldots, \tilde{x}_{tc}^k\}$ is the kth sequence with c segments, input to the lower layer CRFs, and $\tilde{\mathbf{y}}_t^k = \{\tilde{y}_{t1}^k, \ldots, \tilde{y}_{tc}^k\}$ denotes the corresponding output phoneme label sequence. $\tilde{x}_{tj}^k$ denotes the jth input vector of the respective component, having subphone labels as its elements. We use a data driven clustering procedure described below to define subphone and phoneme labels.

To define subphone labels, each $\tilde{H}_j, \tilde{M}_j, \tilde{O}_j, \tilde{L}_j$ segment is divided into $M = 10$ equal intervals to yield $\tilde{H}_j = \{\rho_{j1}^h, \rho_{j2}^h, \ldots, \rho_{j10}^h\}$, $\tilde{M}_j = \{\rho_{j1}^m, \rho_{j2}^m, \ldots, \rho_{j10}^m\}$, $\tilde{O}_j = \{\rho_{j1}^o, \rho_{j2}^o, \ldots, \rho_{j10}^o\}$, $\tilde{L}_j = \{\rho_{j1}^l, \rho_{j2}^l, \ldots, \rho_{j10}^l\}$. The procedure described below is then applied to the four components independently to define their respective subphone labels:

(i) Pick N_q sign segments randomly from the entire training data set ($N_q \gg$ the number of signs in the database).
(ii) Divide each segment into $M = 10$ intervals and compute the mean of the vectors within each interval, e.g. for handshape component, the mean vector of the 16-D handshapes in the interval is computed. Thus, the total number of vectors obtained for clustering is $M \times N_q$.
(iii) Use the Euclidean distance as similarity measure to cluster the $M \times N_q$ vectors using the affinity propagation (AP) algorithm of Frey and Dueck [37].
(iv) Use the $\hat{k}$ exemplars found by the AP algorithm to initialize k-means clustering.
(v) The final centroids obtained by k-means clustering are used as the templates for the subphones. A subphone label $\hat{j}$ is given to an interval if its mean vector is closest to the $\hat{j}$th cluster centroid. The collection of $M = 10$ subphone labels obtained for a component segment are used as elements of the input vector $\tilde{x}_{tj}$ to the corresponding CRF.

Phoneme labels for the three static components (handshape, orientation and location) are defined using a procedure similar to that described above for subphones. In these components, only the starting and ending values are useful. Hence, the mean vectors of the first and last intervals are concatenated and only the resulting vector is used, yielding a total of N_q vectors for clustering. For the movement component, the PCA-based phoneme transcription procedure from our previous works [30,33] is used. We first segment the movement trajectories represented by 3-D positions using naïve Bayesian or rule-based classifiers. Geometric descriptors for segments, such as plane, shape, dominant direction, etc., are then extracted based on PCA of the segments. These descriptors can have sub-classes such as "*xy*-plane", "*yz*-plane", etc. for planes; "line", "circle", etc. for shapes, and so on. These sub-classes are discovered by k-means clustering, and are used to assign movement component phoneme labels to each segment.

The state feature vectors for each CRF model are formed from the corresponding $\tilde{x}_{tj}$ and used to train the four phoneme level linear CRF models independently with the conventional training algorithm. In addition, we also extract trigram features from across the segments as well as from within the segments in a sentence. For example, in a sequence of handshape segments in a sentence, $\tilde{\mathbf{H}} = \{\tilde{H}_1, \tilde{H}_2, \tilde{H}_3, \tilde{H}_4, \tilde{H}_5\}$, each segment $\tilde{H}_j$ consists of a sequence of subphones, $\tilde{H}_j = \{\rho_{j1}^h, \rho_{j2}^h, \ldots, \rho_{j10}^h\}$, $j = 1, \ldots, 5$. Due to the causal nature of the decoding algorithm, the trigram features across segments are computed using only the previous segments, i.e. for segment j and subphone location i, the trigram state feature is obtained by using the subphone labels of "$\rho^h_{(j-2)i}\rho^h_{(j-1)i}\rho^h_{ji}$". However, the trigram feature within segments is obtained by using the subphone labels of "$\rho^h_{j(i-1)}\rho^h_{ji}\rho^h_{j(i+1)}$". The transition features are the labels of the adjacent segments.

6.1.2. Training at the sign level

After the phoneme level training, all the component subphone label sequences in the training sentences are input independently to their respective CRF at the lower level and decoded by the conventional CRF decoding algorithm. The decoded output phoneme sequences are used to train the sign level CRF as in conventional CRFs. The training samples are $\mathcal{D}_s = \{\hat{\mathbf{x}}^k, \mathbf{y}^k\}$, $k=1,\ldots, N$, where $\hat{\mathbf{x}}^k = \{\hat{x}^k_1, \ldots, \hat{x}^k_c\}$ is an input observation sequence at the sign level and $\mathbf{y}^k = \{y^k_1, \ldots, y^k_c\}$ is the corresponding output sequence where ${y_j}^k$ is one of the $\mathcal{K}$ sign labels $\{SIGN_\kappa\}$, $\kappa = 1, \ldots, \mathcal{K}$. We define $\hat{x}^k_j = (\tilde{y}^k_{hj}, \tilde{y}^k_{mj}, \tilde{y}^k_{oj}, \tilde{y}^k_{lj}, \tilde{y}^k_{aj})$ as the jth input feature vector of the kth sequence to the sign level CRF. Its elements are phoneme label outputs from the four parallel CRFs and a segment arc length label, ${y_{aj}}^k$. The latter is obtained by quantizing arc lengths of the movement channel segments by simple thresholding. For sign level recognition, we computed trigram features across segments and used adjacent sign labels as the transition features.

7. Modified segmental decoding algorithm

A given test sentence is automatically segmented, and each sub-segment is labeled as *SIGN* or *ME* (Section 5). Several issues need to be addressed to decode the signs from this pre-processed data. Firstly, the start and end points of the sign segments are unknown and need to be efficiently recovered by proper merging of the sub-segments. Secondly, the test sequence sub-segments will have labeling errors, especially on data from new signers. To address these issues, we developed a new decoding procedure for the sign-level semi-Markov CRF model (Fig. 3) by modifying the decoding algorithm of [31]; however, the standard CRF decoding algorithm is used for the component CRFs at the phoneme level.

7.1. The basic algorithm

The input to the two-layer CRF is data from a segmented test sentence. For simplicity of explanation, we assume that the ME sub-segments have been removed correctly to yield a sequence of sign sub-segments, which if correctly merged into segments, would lead to accurate sign recognition. In our strategy, the sign level directs the formation of test segments by merging adjacent sub-segments, and hypothesizing that they form correct sign segments, selects their labels with a Viterbi-like procedure to compute the most likely sign sequence. Thus, given a hypothesis segment generated dynamically, and its associated normalized feature vectors (Section 4), each segment is divided into $M=10$ equal intervals and from here, the bottom-up flow of information is similar to that during training (each interval is assigned a subphone label, and this collection of labels is decoded by a lower layer CRF into its respective component phoneme; the four components phonemes and the arc length label are decoded by the semi-Markov CRF to assign a sign label to the hypothesized segment). In our algorithm, since sub-segments are merged into segments, the lengths of the final decoded sign label sequence and the input sub-segment sequence will be different; in addition segments are dynamically hypothesized and processed as the decoding proceeds. Due to these differences from the standard semi-Markov CRF model, we modified the standard recursion in (5) for sign level decoding as described below.

We first need to redefine some terms used in the formulation of the standard semi-Markov CRFs. Here, let $\mathbf{S} = \{S_1, S_2, \ldots, S_p\}$ denote a sequence of segments formed by an arbitrary merging of contiguous sub-segments from the sub-segment sequence $\mathbf{s} = \{s_1, \ldots, s_n\}$, and let $\hat{\mathbf{x}}$ be the corresponding input observation sequence consisting of vectors extracted from each $S_i \in \mathbf{S}$. As before, let u_t and v_t in $S_t = (s_{u_t} : s_{v_t}, y_t)$ be the sub-segment positions in $\mathbf{s}$ which describe the start and end positions of S_t and let y_t be its sign label. The inferencing is formulated to find the best sub-segment mergings and their associated labels as

$$\mathbf{S}^* = \underset{\mathbf{S}}{\operatorname{argmax}}\, p(\mathbf{S}|\hat{\mathbf{x}}). \tag{7}$$

Similar to (5), let L be the upper bound on the number of sub-segments that can be merged to form a segment, and let ${}^q\mathbf{S}_{1:r}$ denote the set of all possible partial mergings in $\mathbf{s}' = \{s_1, \ldots, s_r\}$, such that the last segment has label q. Further, let $\delta_r(q)$ denote the largest value of $p(\mathbf{S}'|\hat{\mathbf{x}}')$ for some $\mathbf{S}' \in {}^q\mathbf{S}_{1:r}$ and define $\hat{\mathbf{x}}_{j_1} : \hat{\mathbf{x}}_{j_2}$ as an input observation sequence obtained from sub-segments merged from position j_1 to j_2. The modified recursion is then written as

$$\delta_r(q) = \begin{cases} \max\limits_{\substack{\hat{q} \\ d = 1,\ldots,L}} \delta_{r-d}(\hat{q})\Phi_r(\hat{q}, q, \hat{\mathbf{x}}_{(r-d+1)} : \hat{\mathbf{x}}_r) & \text{if } r > 0, \\ 1 & \text{if } r = 0, \\ 0 & \text{if } r < 0, \end{cases} \tag{8}$$

where

$$\Phi_r(\hat{q}, q, \hat{\mathbf{x}}_{(r-d+1)} : \hat{\mathbf{x}}_r) = \exp\left(\sum_{i=1}^{h} \lambda_i f_i(y_{t-1} = \hat{q}, y_t = q, \hat{\mathbf{x}}_{(r-d+1)} : \hat{\mathbf{x}}_r)\right), \tag{9}$$

and $\hat{\mathbf{x}}_{(r-d+1)} : \hat{\mathbf{x}}_r$ denotes an input observation sequence with merged sub-segments from position $r-d+1$ to position r. Given that $t-1$ signs have been decoded with the last sign having label $\hat{q}$, Φ_r in (8) and (9) represents the cost of the tth merged segment $s_{(r-d+1)} : s_r$ having the sign label q. Though (5) and (8) appear to be similar, the important distinction is that in the latter case the sub-segments are actually merged and feature vectors are then extracted from them. In our formulation, r denotes the position of a sub-segment in the sequence $\mathbf{s}$ and d denotes the length of a hypothesized segment (the number of sub-segments to be merged).

The above scheme for sign level decoding in the two-layer CRF requires component phonemes decoded by the lower layer. If these phoneme sequences are decoded independently in the four channels, different segment lengths may form, leading to higher complexity for sign recognition. Hence, our approach is to allow the decoding process at the phoneme and sign levels to proceed simultaneously, and fuse the decoded component phonemes for sign recognition as the decoding proceeds. In this scheme, the decoding algorithm in (8) and (9) is used only for the sign level CRF and the conventional decoding algorithm is used for the phoneme level CRFs. In the latter, partial phoneme sequences are parallelly decoded and used in (8) and (9) for sign level recognition.

We now describe how the decoding takes place in the two layer CRF by means of an illustrative example. Assume that we have found four sign sub-segments $\mathbf{s} = \{s_1, s_2, s_3, s_4\}$ in a sequence which actually consists of two sign segments as shown in Fig. 4(a) and let $L=2$. Also, let $\mathbf{h} = \{h_1, \ldots, h_4\}$, $m = \{m_1, \ldots, m_4\}$, $o = \{o_1, \ldots, o_4\}$, $l = \{l_1, \ldots, l_4\}$ be the corresponding sub-segments at the component/phoneme level.

(i) For $r=1$, we need to compute $\delta_1(q)$ in (8). Here, the initial conditions for $\delta_{r-d}(\hat{q})$ are taken as 1 and 0, for $d=1$ and $d>1$, respectively. Hence, it is sufficient to consider only the case $d=1$, which gives a non-zero value for $\delta_1(q)$. Evaluating the

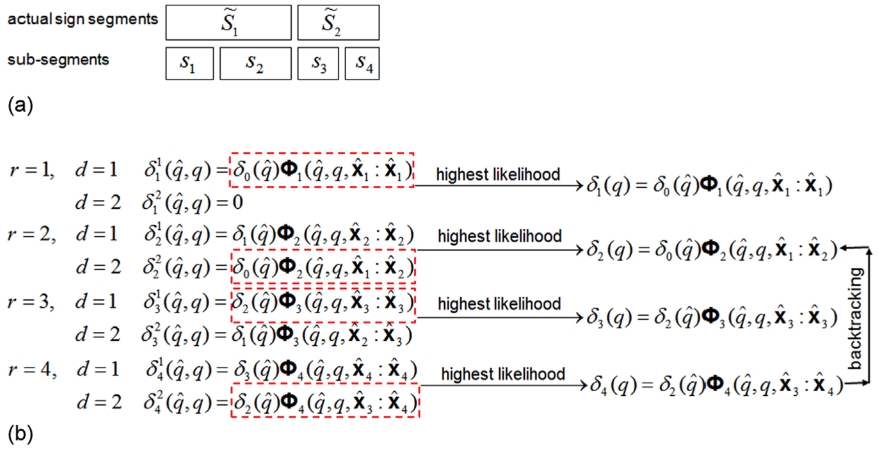
actual sign segments
$\tilde{S}_1$
$\tilde{S}_2$
sub-segments
s_1
s_2
s_3
s_4
(a)
$r=1,\ d=1\quad \delta_1^1(\hat{q},q)=\delta_0(\hat{q})\boldsymbol{\Phi}_1(\hat{q},q,\hat{\mathbf{x}}_1:\hat{\mathbf{x}}_1)$
$d=2\quad \delta_1^2(\hat{q},q)=0$
highest likelihood
$\delta_1(q)=\delta_0(\hat{q})\boldsymbol{\Phi}_1(\hat{q},q,\hat{\mathbf{x}}_1:\hat{\mathbf{x}}_1)$
$r=2,\ d=1\quad \delta_2^1(\hat{q},q)=\delta_1(\hat{q})\boldsymbol{\Phi}_2(\hat{q},q,\hat{\mathbf{x}}_2:\hat{\mathbf{x}}_2)$
$d=2\quad \delta_2^2(\hat{q},q)=\delta_0(\hat{q})\boldsymbol{\Phi}_2(\hat{q},q,\hat{\mathbf{x}}_1:\hat{\mathbf{x}}_2)$
highest likelihood
$\delta_2(q)=\delta_0(\hat{q})\boldsymbol{\Phi}_2(\hat{q},q,\hat{\mathbf{x}}_1:\hat{\mathbf{x}}_2)$
$r=3,\ d=1\quad \delta_3^1(\hat{q},q)=\delta_2(\hat{q})\boldsymbol{\Phi}_3(\hat{q},q,\hat{\mathbf{x}}_3:\hat{\mathbf{x}}_3)$
$d=2\quad \delta_3^2(\hat{q},q)=\delta_1(\hat{q})\boldsymbol{\Phi}_3(\hat{q},q,\hat{\mathbf{x}}_2:\hat{\mathbf{x}}_3)$
highest likelihood
$\delta_3(q)=\delta_2(\hat{q})\boldsymbol{\Phi}_3(\hat{q},q,\hat{\mathbf{x}}_3:\hat{\mathbf{x}}_3)$
$r=4,\ d=1\quad \delta_4^1(\hat{q},q)=\delta_3(\hat{q})\boldsymbol{\Phi}_4(\hat{q},q,\hat{\mathbf{x}}_4:\hat{\mathbf{x}}_4)$
$d=2\quad \delta_4^2(\hat{q},q)=\delta_2(\hat{q})\boldsymbol{\Phi}_4(\hat{q},q,\hat{\mathbf{x}}_3:\hat{\mathbf{x}}_4)$
highest likelihood
$\delta_4(q)=\delta_2(\hat{q})\boldsymbol{\Phi}_4(\hat{q},q,\hat{\mathbf{x}}_3:\hat{\mathbf{x}}_4)$
backtracking
(b)

Fig. 4. An example to illustrate the decoding procedure. (a) A sign sequence and (b) The decoding steps.

other term $\Phi_r(\hat{q},q,\hat{\mathbf{x}}_{(r-d+1)}:\hat{\mathbf{x}}_r)$ in (8) requires the hypothesized sign segment $s_1:s_1$ and the corresponding segments, $\{h_1:h_1, m_1:m_1, o_1:o_1, l_1:l_1\}$ from the component channels. Each component segment is divided into 10 intervals, and a subphone label from the corresponding channel is assigned to each interval. These labels are concatenated into a vector, and input to the corresponding phoneme level CRF to decode the component phonemes using the standard CRF decoding algorithm. The phoneme label outputs from the four CRFs and the arc length label from the movement channel for the hypothesized segment are concatenated to form an input vector $\hat{\mathbf{x}}_1:\hat{\mathbf{x}}_1$ for the semi-Markov CRF at the sign level. This allows $\Phi_1(\cdot)$ to be computed, yielding $\delta_1(q)=\max_{\hat{q}}\Phi_1(\hat{q},q,\hat{\mathbf{x}}_1:\hat{\mathbf{x}}_1)$ for each q. These optimal decoding costs for each sign are stored along with the fact that $d=1$, for use in further decoding steps. For the example considered, this cost is shown highlighted for $d=1$ and some sign q, in Fig. 4(b). In the rightmost part of Fig. 4(b), $\delta_1(q)=\delta_0(\hat{q})\Phi_r(\hat{q},q,\hat{\mathbf{x}}_1:\hat{\mathbf{x}}_1)$, etc. should be read as, "the best cost for decoding a sign q at step $r=1$, is obtained for $d=1$, and a merged segment $s_1:s_1$, from a decoded sign $\hat{q}$ at the previous step". In general, when $r-d<0$, $\delta_{r-d}=0$, so there is no contribution to the decoding cost, and $\Phi_r(\cdot)$ is not computed. When $r-d=0$, $\delta_{r-d}=1$, and the decoding cost is given by $\Phi_r(\cdot)$.

(ii) In general, to compute $\delta_r(q)$ for $r\geq L$, $\Phi_r(\cdot)$ must be evaluated for L new hypothesized sign segments $s_{r-d+1}:s_r$, $d=1,2,\ldots,L$. These values can be computed as outlined in step (i), and weighted by the corresponding stored values $\delta_{r-d}(\hat{q})$ from the previous L decoding steps to yield $\delta_r^d(\hat{q},q)=\delta_{r-d}(\hat{q})\Phi_r(\hat{q},q,\hat{\mathbf{x}}_{(r-d+1)}:\mathbf{x}_r)$, $d=1,2,\ldots,L$; $\hat{q},q=1,2,\ldots,\mathcal{K}$, assuming a $\mathcal{K}$ sign vocabulary. This requires $O(\mathcal{K}^2L)$ computations for each r. Then $\delta_r(q)=\max_{d,\hat{q}}\delta_r^d(\hat{q},q)$ yields the best cost for partially merged segments up to step r, ending in a sign q. For a given sign q, the value of d which maximizes $\delta_r(q)$ specifies the segment for that sign as $s_{r-d+1}:s_r$. For further decoding steps, the best cost $\delta_r(q)$ at this step, and the corresponding values of d and $\hat{q}$ which yield the best cost for each sign must be stored.

For the example in Fig. 4, at step $r=3$, computation of $\delta_3(q)$ requires the sign level sub-segment sequence $\mathbf{s}=\{s_1,s_2,s_3\}$ and the corresponding segments $\mathbf{h}=\{h_1,h_2,h_3\}$, $\mathbf{m}=\{m_1,m_2,m_3\}$, $\mathbf{o}=\{o_1,o_2,o_3\}$, $\mathbf{l}=\{l_1,l_2,l_3\}$ at the phoneme level. To compute $\Phi_3(\cdot)$, two new hypothesis segments that must be considered are $s_3:s_3$ for $d=1$ and $s_2:s_3$ for $d=2$. In the former case, the cost $\delta_3^1(\hat{q},q)=\delta_2(\hat{q})\Phi_3(\hat{q},q,\hat{\mathbf{x}}_3:\hat{\mathbf{x}}_3)$ needs to be evaluated. The values of $\delta_2(\hat{q})$ are available from the previous ($r=2$) decoding step. Also available are the values of d at that step which optimized $\delta_2(\hat{q})$. Suppose this happens for $d=2$ in our example; this is shown highlighted at step $r=2$ in Fig. 4(b). This implies that the best merged segment just preceding the hypothesized segment $s_3:s_3$ is $s_1:s_2$ ($s_{r-d+1}:s_r$ for $r=d=2$). Thus, $\delta_3^1(\hat{q},q)$ as calculated above corresponds to the cost of the merged segment sequence $\{s_1:s_2, s_3:s_3\}$, ending with label q. For $d=2$, the cost is $\delta_3^2(\hat{q},q)=\delta_1(\hat{q})\Phi_3(\hat{q},q,\hat{\mathbf{x}}_2:\hat{\mathbf{x}}_3)$. The hypothesized segment here is $s_2:s_3$ and $\delta_1(\hat{q})$ represents the optimal decoding cost to a sign label $\hat{q}$ at step 1. This is attained for $d=1$ at that step, giving rise to a merged segment sequence $\{s_1:s_1, s_2:s_3\}$ with associated cost $\delta_3^2(\hat{q},q)$. In this example, we assume that the largest cost is $\delta_3^1(\hat{q},q)$ for $d=1$, and it is shown highlighted in Fig. 4(b). At this step, the best merged segment sequence is $\{s_1:s_2, s_2:s_3\}$ ending in a sign label q. Here, the relevant information – the values of $\delta_3(q)$, and $d,\hat{q}$ which optimize δ_3 for each q are stored for use in further decoding steps.

(iii) On reaching the end of the sub-segment sequence, the optimal sign sequence and corresponding merged segments are retrieved by backtracking. At step $r=4$ in Fig. 4(b), assume that the best value $\delta_4(q)$ over all q is obtained for $d=2$, and some $\hat{q},q$, i.e. $\delta_4^2(\hat{q},q)=\delta_2(\hat{q})\Phi_4(\hat{q},q,\hat{\mathbf{x}}_3:\hat{\mathbf{x}}_4)$. Here the hypothesized segment is $s_3:s_4$ and the best cost of the segment sequence preceding it is $\delta_2(\hat{q})$, i.e. a segment sequence formed at step 2. Hence, backtracking to step 2, it is found that best cost there occurs for $d=2$ and some $\hat{q}$, with the merged segment $s_1:s_2$. Thus the decoding finally yields the merged segment sequence $\{s_1:s_2, s_3:s_4\}$ and their associated sign labels.

7.2. Two-class SVMs

The modified decoding algorithm described in Section 7.1 makes a forced assignment of one of the sign labels (which has the best cost) to a hypothesized segment; this may not always be appropriate. For example, test sentences may have misclassified *SIGN* and *ME* sub-segments, and hypothesized segments may actually contain ME segments which the decoding algorithm has not been trained to recognize. Hence, it is desirable to have a mechanism to indicate to the decoder whether a hypothesized segment is likely to correspond to one of the valid signs or not.

Our approach is to use two-class SVMs to discriminate between valid or invalid sign segments. This reduces the complexity of the problem significantly from a large multi-class problem to a set of two-class problems. In this case, addition of new signs will not affect the trained SVMs and only one new SVM needs to be trained for each new sign. The main task is to generate examples of invalid samples; the positive examples can be obtained by using the sign segments from a particular class. We used a simple approach to generate a sufficient number of negative examples for each sign, as follows. In any training sentence (both, with and without ME sub-segments), starting from the end of a correct sign segment, form new segments, containing up to L contiguous sub-segments. In this construction, it is known a priori which of these segments form a valid sign, and which do not. Now, use the conventional linear CRF to obtain a sign label for each of the incorrectly merged segments. The segment is then taken as an *INVALID* instance of the obtained sign label. This step is repeated for all training sequences. There is a possibility that no negative samples will be generated for some signs in which case, these signs need to be handled separately. However, this problem did not occur in our data set.

All the generated negative examples for a sign are used with its positive examples to train a two-class SVM. As these SVMs are incorporated into the two-layer CRF model, SVMs with probabilistic outputs as described in [34], are used. The features used for training these SVMs are similar to the features used for training these *SIGN*/*ME* SVM classifier of Section 5 and are listed in the SVM2 column of Table A1 (Appendix). Here, the only difference is that we do not use the transition features described in Section 5 as the two-class SVMs need to discriminate using only within segment features. The features listed in Table A1 are concatenated to form 126-D real-valued feature vectors for input to the two-class SVMs.

The two-class SVMs are integrated into the decoding algorithm by introducing additional steps for maximizing the likelihood. Instead of directly obtaining the maximum and making a forced choice for the corresponding sign label, the most likely sign label for each hypothesized segment is first determined, and the two-class SVM for this sign label is used to check if the hypothesized segment is valid. Only the valid hypothesized segments are used for final decoding. In case all hypothesized segments at a particular step r are declared as invalid by the two-class SVMs, we fall back on the default decoding procedure without SVMs. We denote this additional functionality as "svmmax()" in the final decoding algorithm given in (10), below.

7.3. Modified decoding algorithm with skip states

The *SIGN* and *ME* labels of the sub-segments provided by the classifier of Section 5 can also be used to break the complete sequence into partial sequences, i.e. the positions of the *ME* sub-segments which have been discarded can be used as indicators of potential boundary points across which merging is not necessary. Hence, when an *ME* label is encountered, the sequence is broken into two independent partial sequences and sub-segments from the two partial sequences are never merged. This reduces the decoding computations.

As the *SIGN*/*ME* classifier described in Section 5 is not error free, the final test sequences may include sub-segments which are actually ME and/or be missing some of the actual sign sub-segments which may have been erroneously classified as *ME* and discarded. The decoding algorithm described in Section 7.1 assumed for simplicity that sub-segment classification was perfect, so that sub-segments were merged with their immediate neighbors without the need for skipping any of the sub-segments. However, if a sequence consists of *ME* sub-segments erroneously labeled as *SIGN*, it would be desirable to infer this and skip these sub-segments. To facilitate this, we modified the decoding algorithm as follows. Let M_s be the maximum number of sub-segments that can be skipped. Together with the inclusion of the two-class SVMs in the decoding algorithm, the recursive formulation of (8) and (9) is modified as

$$\delta_r(q) = \begin{cases} \underset{\substack{\hat{q},\ d=1,\dots,L,\\ t_s=0,\dots,M_s}}{\text{svmmax}} \delta_{r-d-t_s}(\hat{q}) \Phi_r(\hat{q}, q, \hat{\mathbf{x}}_{(r-d+1)} : \hat{\mathbf{x}}_r) & \text{if } r>0, \\ 1 & \text{if } r=0, \\ 0 & \text{if } r<0, \end{cases} \tag{10}$$

where t_s denotes the number of sub-segments to be skipped.

Suppose $\mathbf{s} = \{s_1, s_2, s_3\}$ is a test sequence with a classification error only in sub-segment 2 (i.e. it is actually not *SIGN*, but *ME*). If no skip state is allowed, s_2 has to be included to evaluate the most likely path among three possible segment sequences $\mathbf{S}_1 = \{s_1, s_2, s_3\}$, $\mathbf{S}_2 = \{s_1 : s_2, s_3\}$ and $\mathbf{S}_3 = \{s_1, s_2 : s_3\}$, despite the error. In the extended decoding algorithm with skip state, if s_2 is skipped (assuming that it is ME), then it would demarcate two sub-sequences, one ending in s_1, and the other beginning with s_3, as these two sub-segments will not be merged across ME sub-segments. The complete modified decoding algorithm is given below:

(i) Given a sequence of sub-segments $\mathbf{s} = \{s_1, \dots, s_n\}$, with s_j classified as *SIGN* or *ME*.
(ii) Compute the recursive term in (10) for all partial sequences with a suitable choice of $M_s > 0$. Increasing M_s increases the computational cost. When an *ME* label is encountered, the search ends and further sub-segment merging is stopped. The next partial sequence is treated as a new sequence and merging is only done with succeeding sub-segments.
(iii) The recursive computations are continued until the end of the sequence or until an *ME* label is encountered; backtracking then retrieves the sign sequence.

7.4. Computational complexity

The inferencing computation for semi-Markov CRFs is more expensive than conventional linear-CRFs as it needs to consider several potential segment lengths d and skip states M_s, when optimizing the cost. As described in Section 7.1, the decoding cost per sub-segment without skip states is $O(\mathcal{K}^2 L)$. If up to M_s skip states are included, then this cost increases to $O(\mathcal{K}^2 L(M_s+1))$. Thus the additional computational complexity over linear CRFs is only linear in L and M_s. In our experiments, the maximum number of sub-segments in a sign segment was found to be seven, thus we used $L=7$. After discarding the *ME* sub-segments using the *SIGN*/*ME* classifier, the maximum number of consecutive *ME* sub-segments that are erroneously classified as *SIGN* in a hypothesized sign segment was experimentally found to be about two, hence, we chose $M_s=2$. Compared to linear-CRFs and semi-Markov CRFs, the increase in computational cost incurred by using the two-class SVMs (to classify segments as valid or invalid signs) and extracting features on the fly along with the inferencing procedure is minor.

8. Experimental results and discussions

We present detailed experimental results for (1) the *SIGN*/*ME* sub-segment classifier (subsystem I), and (2) the two-layer CRF recognition framework (subsystem II), used to recognize naturally signed continuous ASL sentences.

8.1. Data collection for continuous ASL

We used a CyberGlove [38] and Polhemus FASTRACK system [39] to acquire the handshape, palm orientation and hand position data. The tracker and glove data were synchronized at a frame rate of 31.10 ms. Two trackers were attached to the back of each signer's hands and a third to the lower back, to serve as a reference for position and orientation of the signer relative to the transmitter. Conceptually, each sensor has an attached orthogonal coordinate frame, and we calculated the relative readings for the position and orientation of the hands as in [40]. The raw handshape, palm orientation and location data are 16-D, 9-D and 3-D vectors, respectively. All experiments were conducted using the four components from the right hand, obtained from tracker and glove data. For each sentence, a corresponding video sequence of the frontal view of the signer was also recorded. This served as a useful visual aid during manual processing of glove and tracker data to generate appropriate training sets for various experiments.

The data for the continuous sign recognition experiments was obtained from eight signers including seven deaf persons and one hearing person. The seven deaf subjects were native signers of the local sign language and the last was an expert signer. The signed sentences were performed continuously, without pauses between signs and closely followed ASL grammar. There were in total 74 distinct sentences from a 107-sign vocabulary that included basic signs and signs with directional verb inflections. Each sentence was made up of 2–6 signs. The average number of samples per signer for each distinct sentence was between 3 and 10, providing a total of 2393 sentences and 10 852 sign instances. The training and testing was done using a full round robin procedure by leaving one signer out as an unseen signer (i.e. signer whose data was not used for training) each time. We also used 80% of the data from seven signers for training and the remaining 20% as unseen samples from seen signers for testing.

8.2. Subsystem I

We first trained a CRF and a SVM to label the sub-segments and investigated their performance independently, followed by the BN fusion scheme. Based on the extracted state and transition features, we trained a linear-chain CRF, using the features listed in Table A1. The important settings for training and testing the CRF are summarized in Table 1.

The original continuous-valued features extracted from handshape, movement, orientation and location components were converted to discrete symbols by k-means clustering before being used as inputs to CRFs. We used only the training data to find the clusters, and the trained centroids were labeled as $1, 2, \ldots, \mathcal{M}$. For test data, the feature vector was assigned a number corresponding to its nearest cluster centroid. To determine the optimal number of clusters $\hat{k}$ for each feature, we searched in a range of potential values based on the number specified by linguists. For every k, we trained a CRF to classify the sub-segments as *SIGN* or *ME* based only on the concerned feature, and selected the $\hat{k}=k$ which yielded the best accuracy. For example, if there were about 40 handshapes defined by linguists, we searched for $\hat{k}$ in the range of 30–80 in steps of 5 or 10 for the state or transition features related to handshape. The $\hat{k}$ values for most of the features were in the range of 50–80 except for arc length features which had $\hat{k}=10$. The average classification accuracy obtained from individual features was 65.2%.

Table 1
Settings used for CRFs.

Setting	Description
Output labels	Two class: *SIGN* and *ME*
No. of state features	25
No. of transition features	6
Optimization	Quasi-Newton algorithm (LBFGS)
Regularization	L_2-norm
Decoding	Viterbi algorithm

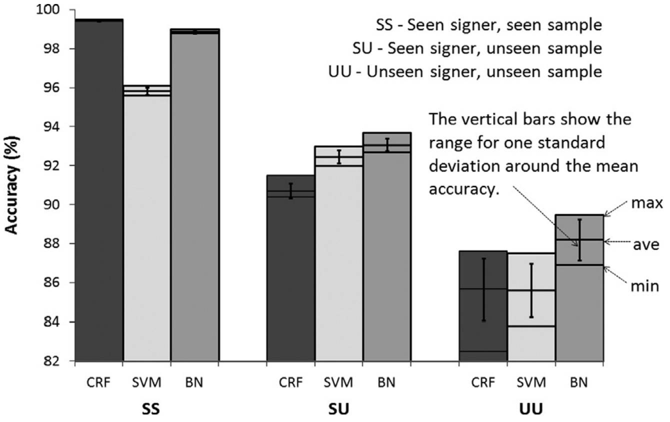

Fig. 5. *SIGN* and *ME* classification accuracy of CRF/SVM/BN. The diagram shows the maximum, minimum, average values and the range for one standard deviation from the mean.

Fig. 5 summarizes the classification results of the CRF trained using all features with their optimal $\hat{k}$ values. A good value of C (regularization parameter (3)) was determined experimentally as $C=0.085$; however, the classification results were not very sensitive to C. The accuracy was consistently good for unseen samples from seen signers (SU samples) and averaged 90.7 ± 0.4% (the variation from the mean that is indicated here and in all accuracy results refers one standard deviation), while for unseen signers (UU samples), it was 85.7 ± 1.6%. This shows that the trained CRFs can generalize quite well to data from new signers.

For the SVM classifier, the features listed in Table A1 were concatenated to form 126-D continuous-valued feature vectors. The elements of the feature vectors were normalized to have zero mean and unit variance. We used Gaussian radial basis functions as the SVM kernels, with the regularization parameter set to 5, as found by experimental tuning. The classification results with the SVM are shown in Fig. 5.

The SVM performed about 2% better than the CRF classifier for SU samples. For UU samples, consistent classification accuracies were obtained for all eight rounds with CRFs and SVMs yielding average accuracies of 85.7 ± 1.6% and 85.6 ± 1.4%, respectively. Examination of the results showed that the errors made by both approaches were different although they showed similar classification accuracies. Hence, we fused the outputs of the two classifiers through a BN to improve performance.

We repeated the experiment by using the BN shown in Fig. 2. The real-valued features were quantized as described in Table A2, and the conditional probability table (CPT) of the discrete nodes was learned by maximum likelihood estimation. During testing, the nodes **crfErr**, **svmErr** and **fLabel** were inferred based on the input observations. The final BN classification accuracy shown in Fig. 5 is improved compared to individual CRF and SVM results, especially for UU samples where the BN gave an average improvement of about 2.5% in accuracy compared to both CRF and SVM. Accuracy for unseen signers is important for the later part of our work when we need to incorporate the classified *SIGN* and *ME* labels into the final recognition scheme.

8.3. Subsystem II

The component phonemes were obtained from training data for each round of the experiments, using the transcription procedure of Section 6. However, for the movement component the phonemes were obtained by using the transcription procedure with PCA-based representations [30,33]. For the other three components, we randomly selected 5000 segments

Table 2
Settings used for training phoneme and sign level CRFs.

Setting	Phoneme level	Sign level
Output label	The phonemes defined for each component by the transcription procedure	107 signs
No. of state features	30	10
No. of transition features	1	1
Optimization	Quasi-newton algorithm (LBFGS)	Quasi-newton algorithm (LBFGS)
Regularization	L_2-norm	L_2-norm
Decoding	Viterbi algorithm.	Viterbi algorithm

(i.e. $N_p = 5000$) for AP clustering. Features from the starting and ending intervals of each segment were concatenated, resulting in 5000 feature vectors for AP clustering. The “preference” parameter in the AP algorithm which affects the number of phonemes obtained was set to $\rho\tilde{s}_{min}$, where $\tilde{s}_{min}$ is the minimum value of pairwise data point similarities based on Euclidean distance, and ρ is a scaling constant for parameter tuning. The AP clustering algorithm was run with different scale factors for 10 sets of 5000 randomly chosen segments to find the best number of clusters for representing the phonemes. We used the exemplars obtained from AP clustering to initialize the k-means algorithm for final clustering. The scale factors which led to the smallest errors from k-means were experimentally found to be 1.7, 4.5 and 1.3 for handshape, orientation and location, respectively.

The subphones for each component were extracted by using the method described in Section 6.1.1 which is similar to the phoneme extraction procedure for the static components. We randomly selected 1000 segments ($N_q = 1000$) and clustered them by AP. Each segment was divided into 10 intervals and the mean feature vector was obtained for each interval. Thus, the total number of feature vectors for clustering the subphones using AP was 10 000. Following the procedure for phoneme definition, scale factors of 1.25, 1.00, 2.00, 0.30 were used for handshape, movement, orientation and location, respectively. The “best” number of phonemes and subphones was obtained experimentally for each round, and was found to be comparable to those defined by linguists. The average number of phonemes and subphones was 32, 39, 31, 27, and 40, 50, 41, 45 in the *H*, *M*, *O* and *L* channels, respectively.

8.3.1. Sign versus non-sign classification by SVM

The procedure of Section 7.2 was used to generate the $SIGN_\kappa$, $\kappa = 1, \ldots, 107$ and *INVALID* training segments from the training sentences. The features listed in Table A1 were concatenated to form 126-D continuous-valued feature vectors for input to two-class SVMs with Gaussian radial basis function kernels. The elements of the feature vectors were normalized to zero mean and unit variance. The SVMs provided probability outputs and a threshold of 0.5 was used to differentiate the $SIGN_\kappa$, $\kappa \in \{1, \ldots, 107\}$ and *INVALID* classes. Different cost functions were tuned experimentally for each SVM.

In the eight rounds of experiments, average accuracies of 94.8% (SU samples) and 93.4% (UU samples) were obtained. Though these test samples are not the actual pool of segments that may be formed during the decoding procedure, it is a subset of the possible samples, and the accuracies can be considered as a promising indicative result. These SVMs were integrated into the decoding algorithm and their functionality was further verified in the recognition experiments.

8.3.2. Continuous sign recognition results

Due to the supervised training of CRFs, we need to address word order variations in sentences, i.e. test sentences which have different word orders from training sentences. To deal with this type of variation, we identified pairs of signs that tended to be swapped in order and used the same weights for the sign label transition features of these pairs so that transitions from one to the other were equally likely. This strategy was applied to the sign level CRF only. We conducted several progressive experiments to evaluate the two-layer CRF-based classifier. The settings for training the phoneme level and sign level CRFs are summarized in Table 2. The recognition performance of the continuously signed sentences was evaluated based on substitution, deletion and insertion errors. We used “recall” and “precision” to measure classifier performance, computed as

$$\text{Recall (Rec)} = \frac{N_c}{N_c + N_s + N_d}, \quad \text{Precision (Pre)} = \frac{N_c}{N_c + N_s + N_i}, \tag{11}$$

where N_c is the number of correct classifications and N_s, N_d, N_i are the number of substitution, deletion and insertion errors, respectively.

For a basic comparison with our CRF-based method, we used standard left to right HMMs. The observation sequences to train the HMMs were obtained by concatenating the normalized vectors (described in Section 4) from all the components to form 31-D feature vectors. We modeled each sign with 3–5 states and each state was represented by a single Gaussian with full covariance matrix. The Viterbi algorithm was used to decode the sign sequences. For training, the transition probabilities were set to be equi-probable except for the invalid transitions, whose probabilities were set to zero. We divided the sign segments from the training data into 3–5 sub-segments with equal arc length and used them to initialize the Gaussian parameters in each state. We attempted to use the same approach used in the CRF-based framework to tackle the word order issue in the sentences. However, the performance was not as good as the naturally trained parameters. Hence, we did not adjust the HMM parameters.

(A) *Clean sign segment recognition*: To systematically evaluate the two-layer CRF framework, we first checked performance by using manually specified boundary points of the segments in test sequences, i.e. a sequence of isolated sign segments, obtained after discarding the ME segments. The features were extracted directly from these sign segments. The decoding procedure is straightforward in this case. The component phonemes were decoded independently in the four parallel channels and together with the arc length labels of the sign segments, the phoneme label outputs of the four components were concatenated at every time instant. These were input to the sign level CRF and the standard CRF decoding algorithm was applied to obtain the sign sequences. We found that the recognition accuracies were not very sensitive to the regularizing parameter C (3), though there was a slight drop in accuracy beyond the range $0.1 < C < 0.8$. Hence, C was chosen in this range for training both phoneme and sign level CRFs. The average recognition accuracy for the test sentences in the eight round robin experiments consisting of sequences of clean

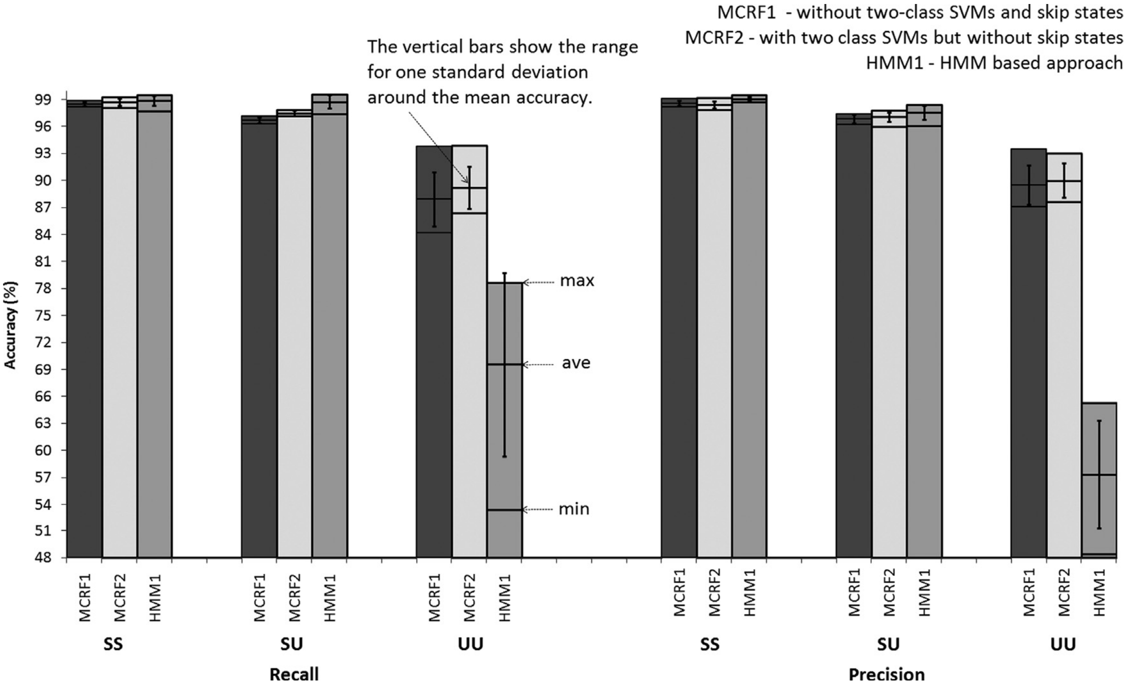

Fig. 6. Recognition accuracy with segmental decoding procedure. The diagram shows the maximum, minimum, average values and the range for one standard deviation from the mean.

segments was 98.0 ± 0.4% for SU samples (minimum 97.4%) and 90.8 ± 2.7% for UU samples (minimum 88.0%) indicating good performance and good generalization. These results provide an experimental upper bound for the accuracy of our CRF-based classifier that would be obtained if the *SIGN*/*ME* labeling was perfect, and the *SIGN* sub-segments were perfectly merged.

(B) *Recognition by merging sign sub-segments*: The next set of experiments considered continuously signed sentences which were automatically segmented, and ME sub-segments in all sentences were manually discarded. This left only over-segmented signs in the testing sentences with unknown segment boundary points. We conducted three experiments to verify the proposed decoding algorithm described in Sections 7.1 and 7.2. As the sign sentences do not contain ME, skip states are not required; the task for the decoding algorithm is to merge the sub-segments and recognize the sign sequences correctly. The first two experiments were based on CRFs ($L=7$ was used) while the third used HMMs (trained as described in Section 8.3.2). For the first experiment, we used the modified segmental decoding procedure for the two-layer CRF without two-class SVMs (MCRF1) while two-class SVMs (for recognizing *VALID*/*INVALID* sign segments) were used in the decoding procedure (MCRF2) for the second experiment. MCRF2 results can be taken as a measure of performance of the segmental decoding algorithm with the two-class SVMs assuming that the BN had perfectly classified the sub-segments as *SIGN*/*ME*.

The recognition results of MCRF1 and MCRF2 shown in Fig. 6 are comparable. The latter performed slightly better and showed about 1% improvement on the average recall rates for SU and UU samples. The average recall rates for MCRF2 were 97.5 ± 0.3% and 89.2 ± 2.3% for SU and UU samples, respectively. These are close to the clean segment recognition accuracies of 98.0 ± 0.4% and 90.8 ± 2.7%, respectively. HMM1 in Fig. 6 shows the recognition performance with the standard HMM approach. The recognition performance is very good for SU samples and even slightly outperforms our proposed framework. However, when the HMM-based framework was tested with UU samples, performance dropped drastically, yielding an average recall rate of 69.6 ± 10.2% and precision of 57.3 ± 6.0%. Our two-layer CRF-based method outperformed it in this case by 19.6% and 32.7% for recall and precision, respectively. This shows that the generative HMM models do not generalize as well to data from unseen signers.

(C) *Recognition of sentences with movement epenthesis*: Thus far, all experiments considered recognizing signs from continuously signed sentences after manually discarding ME segments. This set of experiments was to evaluate performance on the complete problem where ME sub-segments may be present in the test sequences due to automatic segmentation and sub-segment labeling errors. Here, the complete decoding algorithm with all the proposed features is used.

For our two-layer CRF-based strategy, we used the labeled sub-segment sequences obtained from the BN classifier. We discarded the sub-segments automatically labeled as *ME* and used the remaining sub-segments for merging. We used our modified segmental CRF decoding algorithm without and with the two-class SVMs and skip states (called MCRF3 and MCRF4, respectively). For comparison, we also used the previously trained HMMs based on only sign data to decode the sequences (called HMM2). Here, the inputs to HMM2 was sign data remaining after *ME* sub-segments were discarded. Fig. 7 compares the performance of MCRF3, MCRF4 and HMM2. The results show that the CRF models and HMM perform well for SU samples. This is not surprising as the average classification accuracy of the sub-segment classifier for SU samples was high and we can expect that the recognition accuracies will not deviate appreciably from the experiments where the ME sub-segments were manually discarded (compare with Fig. 6). However, for UU samples, comparing MCRF1 and MCRF3, performance decreased from 87.9 ± 3.0% to 82.8 ± 3.4% and 89.5 ± 2.2% to 86.7 ± 2.5% for recall and precision, respectively. In comparison, in the HMM experiments (HMM1 and HMM2), the recall rate dropped from 69.6 ± 10.2% to 67.6 ± 10.2% and the precision dropped from 57.3 ± 6.0% to 56.4 ± 6.0%.

We experimentally tested MCRF4, the complete two-layer CRF decoding scheme augmented with two-class SVMs and skip states, for $M_s=1$ and 2 (the maximum number of skip states), and found that both yielded comparable results. Hence, we chose $M_s=1$, for less computational cost. The performance of MCRF4 is also shown

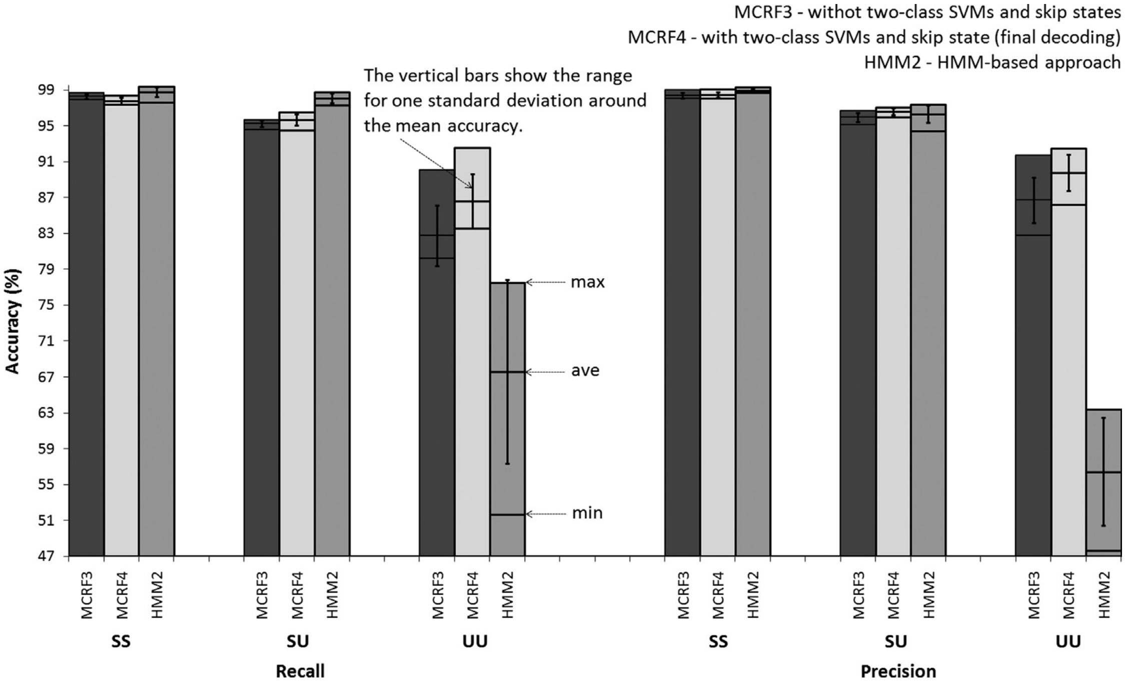

Fig. 7. Recognition accuracy with our CRF-based approach and HMMs. The diagram shows the maximum, minimum, average values and the range for one standard deviation from the mean.

in Fig. 7. A recall rate and precision of 95.7 ± 0.6% and 96.6 ± 0.4% was obtained for SU samples, while 86.6 ± 3.0% and 89.8 ± 2.0%, respectively, was obtained for UU samples. These results are close to the case where ME segments were manually discarded and only merging of sign sub-segments was necessary for recognition (see MCRF2 in Fig. 6); the corresponding MCRF2 results were 97.5 ± 0.3%, 97.1 ± 0.5% (for SU samples) and 89.2 ± 2.3%, 90.0 ± 1.9% (for UU samples). The results from our final decoding algorithm MCRF4 in a realistic sign recognition scenario also compare favorably with the experimental upper bound for recognition accuracy obtained with clean sign segments (98.0 ± 0.4% for SU samples and 90.8 ± 2.7% for UU samples).

We used statistical significance tests to evaluate MCRF4, our proposed classifier versus the HMM2 classifier. For this we used the precision and recall results on UU test data (i.e. data from a signer not used in training) from each of the eight round robin experiments. The null hypothesis H_0 was that the average performance of MCRF4 was at best the same as HMM2, and the alternative hypothesis H_1 was that the average performance of MCRF4 exceeded that of HMM2. Using a paired t-test for precision as well as recall, it was found that the null hypothesis for both measures had very low p-values, below 0.001, calling for rejection of the null hypotheses. The one-sided 95% confidence interval estimates indicate that the average recall and precision rates of MCRF4 exceed that of HMM2 by at least 13.0% and 28.8%, respectively. These experimental results demonstrate that our CRF-based framework can handle signer variations well, and shows good generalization to new signers. Compared to many of the signer independent systems with or without adaptation surveyed in the literature, we have obtained high recognition accuracy for continuously signed sentences by unseen signers.

9. Conclusions

We have devised a segment-based sign recognition approach that is robust to variations which arise in continuous sign language sentences. Moving away from the generative HMM-based recognition approach, we have demonstrated that the discriminative CRF model is better able to deal with variations in sign language and provides good generalization.

Our framework was designed to tolerate variations in continuous sign recognition. We demonstrated the viability of using CRFs in the parallel channels which is commonly modeled with HMMs. The multichannel scheme allows variations in the components to be tackled independently and more efficiently. We proposed a two-layer CRF structure which allows component level and sign level variations to be handled separately for phoneme and sign recognition. All these characteristics contribute to the good recognition performance obtained in our experiments for recognizing continuously signed sentences by unseen signers. In addition, the parameters of the parallel phoneme level CRFs as well as the sign level CRF are learned independently, making the training tractable and fast.

We also devised a novel and efficient decoding algorithm for the two-layer CRF by modifying the semi-Markov CRF inferencing algorithm. We proposed a method to include an *INVALID* class in the CRF-based framework by using two-class SVMs; this increases the robustness of the decoder. We also introduced skip states into the decoding algorithm to handle possible errors by the sub-segment classification algorithm, i.e. ME segments that may have been incorrectly labeled as *SIGN*. Unlike many layered models where decoding is performed separately in different channels or levels, producing either top-down or bottom-up information flow, the data streams at different channels and levels of our proposed framework are modeled simultaneously. The final inferencing results are obtained based on instantaneous fusion of information from the phoneme and sign levels and this leads to natural synchronization between the data streams. Our CRF-based decoding framework has yielded high recognition accuracy for decoding continuously signed sentences by unseen signers, yielding 86.6% recall and 89.8% precision rates. These high rates have been achieved without any adaptation procedure for new signers.

Our conjecture is that modeling movement epentheses is not a good idea for developing signer independent systems as they may include large variations that are not systematic. Hence, in our

Table A1
Features extracted for different classifiers.

Feature	# Sym	Description	CRF	SVM1	SVM2
hand_start	70	Starting handshape	✓	✓	✓
hand_end	70	Ending handshape	✓	✓	✓
hand_msdif	70	Mean of the adjacent handshape differences	✓	✓	✓
diff_strhand[a]	70	Difference of the previous end handshape and the current start handshape	✓	✓	✗
hand_std	–	Handshape standard deviation	✗	✗	✓
orien_start	50	Starting palm orientation	✓	✓	✓
orien_end	50	Ending palm orientation	✓	✓	✓
orien_msdif	80	Mean of the adjacent palm orientation differences	✓	✓	✓
diff_strorien[a]	80	Difference of the previous end palm orientation and the current start palm orientation	✓	✓	✗
orien_std	–	Palm orientation standard deviation	✗	✗	✓
loc_mean	50	Mean of the hand positions	✓	✓	✓
loc_start	50	Starting hand position	✓	✓	✓
loc_end	50	Ending hand position	✓	✓	✓
diff_mloc[a]	60	Difference of the mean of the hand positions of current and previous sub-segments	✓	✓	✗
loc_std	–	Location standard deviation	✗	✗	✓
mov_dom	60	Dominant direction of hand motion	✓	✓	✓
mov_start	60	Starting direction of hand motion	✓	✓	✓
mov_end	60	Ending direction of hand motion	✓	✓	✓
diff_mdom[a]	70	Difference of the mean of the dominant direction of hand motion of current and previous sub-segments	✓	✓	✗
mov_std	–	Movement standard deviation	✗	✗	✓
arc_length	10	Arc length	✓	✓	✓
comb_arc[a]	10	Combined arc length of adjacent sub-segments	✓	✓	✗
label[a]	2	*SIGN* or *ME*	✓	✗	✗
tri_features	–	Trigram features	✓	✗	✗
num_seg	–	Number of sub-segments merged	✗	✗	✓

Note: Features for CRF and SVM1 are based on each sub-segment and features for SVM2 are based on each sign segment.

[a] CRF: transition features and the others are state features; SVM1: *SIGN*/*ME* classifier; SVM2: two-class SVMs.

Table A2
Summary of the BN.

Node	State	Description
fLabel	*SIGN*, *ME*	Sign or movement epenthesis sub-segment
svmErr	Yes, no	Detection by SVM is an error or not
crfErr	Yes, no	Detection by CRF is an error or not
svmProb	1–10	Quantized SVM output probabilities
svmLab	*SIGN*, *ME*	Label from SVM
svmPos	1–4	Position of the sub-segment from SVM classifications
crfProb	1–10	Quantized CRF output probabilities
crfLab	*SIGN*, *ME*	Label from CRF
crfPos	1–4	Position of the sub-segment from CRF classifications
arcLen	1–10	Quantized arc length

approach, ME is differentiated from sign, and discarded from test sentences, before sign decoding. For this, we identified the locations of movement epentheses in a sentence by training a classifier (a BN which fused outputs from a SVM and a CRF) to label *SIGN* and *ME* sub-segments. Any resulting errors are dealt with dynamically during decoding. Further, as opposed to the conventional perception of ME as merely a non-informative transition segment from one sign to the next, we have extracted information that is useful to the decoder. Specifically, the locations of *ME* in a sentence help to break down a sentence into smaller sequences, and thereby reduce the decoding complexity.

Though four manual components are used in sign language communication, most of the previous works do not model them fully in four separate channels. Our recognition framework closely follows the sign language model defined by linguists, and includes all the four components, viz. handshape, movement, palm orientation and location. Modeling the movement component as a separate channel is challenging as it requires features that can represent the trajectory shape and direction of hand motion while being invariant to location and size of the trajectory. For this, we proposed a simple and efficient line fitting procedure [32] to extract features for the movement component in continuously signed sentences, and demonstrated that these features are effective for continuous sign recognition.

In future work, it would be desirable to investigate if the accuracy of the BN classifier for *SIGN*/*ME* labeling can be enhanced by using more efficient features. Scalability to larger vocabularies needs to be explored. Since in our approach, sign recognition is based on phonemes in the four parallel component channels, it has potential to scale to larger vocabularies. However, as more manual signs are added, we may expect that some signs may be close in appearance, in which case additional information would need to be used to discriminate between similar signs. It would also be useful to include data from both hands, grammatical inflections and the non-manual aspects of sign language.

Conflict of interest

None declared.

Acknowledgement

We are grateful to members of the Deaf & Hard-of-Hearing Federation (Singapore) for providing the signing data.

Appendix A

(See Tables A1 and A2).

References

[1] R. Battison, H. Markowicz, J. Woodward, A good rule of thumb: variable phonology in American sign language, in: R.W. Fasold, R.W. Shuy (Eds.), Analyzing Variation in Language, Georgetown University Press, 1975, pp. 291–302.

[2] C. Lucas, R. Bayley, C. Valli, What's your Sign for Pizza?: An Introduction to Variation in American Sign Language, Gallaudet University Press, 2003.

[3] D.M. Perlmutter, On the segmental representation of transitional and bidirectional movements in ASL phonology, in: S.D. Fischer, P. Siple (Eds.), Theoretical Issues in Sign Language Research, vol. 1, The University of Chicago Press, 1990, pp. 67–80.

[4] O. Aran, L. Akarun, A multi-class classification strategy for fisher scores: application to signer independent sign language recognition, Pattern Recognition 43 (2010) 1776–1788.

[5] R. Yang, S. Sarkar, B. Loeding, Enhanced level building algorithm for the movement epenthesis problem in sign language recognition, in: Proceedings of Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007, pp. 1–8.

[6] R. Yang, S. Sarkar, B. Loeding, Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (3) (2010) 462–477.

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

[8] H.-I. Suk, S.-S. Cho, H.-D. Yang, M.-C. Roh, S.-W. Lee, Real-time human–robot interaction based on continuous gesture spotting and recognition, in: Proceedings of International Symposium on Robotics, Seoul, Korea, 2008, pp. 120–123.

[9] H.-D. Yang, S. Sclaroff, S.-W. Lee, Garbage model formulation with conditional random fields for sign language spotting, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 2008.

[10] 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 (7) (2009) 1264–1277.

[11] H.-D. Yang, S.-W. Lee, Robust sign language recognition with hierarchical conditional random fields, in: Proceedings of International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 2202–2205.

[12] H.-D. Yang, S.-W. Lee, Simultaneous spotting of signs and fingerspellings based on hierarchical conditional random fields and boostmap embeddings, Pattern Recognition 43 (1) (2010) 2858–2870.

[13] D. Kelly, J. McDonald, C. Markham, Continuous recognition of motion based gestures in sign language, in: Proceedings of International Conference on Computer Vision Workshops (ICCV Workshops), Kyoto, Japan, 2009, pp. 1073–1080.

[14] D. Kelly, J. McDonald, C. Markham, Recognizing spatiotemporal gestures and movement epenthesis in sign language, in: Proceedings of International Conference on Machine Vision and Image Processing, Dublin, Ireland, 2009, pp. 145–150.

[15] H.-K. Lee, J.H. Kim, An HMM-based threshold model approach for gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (10) (1999) 961–973.

[16] C. Wang, W. Gao, Z. Xuan, A real-time large vocabulary continuous recognition system for Chinese sign language, in: Pacific Rim Conference on Multimedia, Beijing, China, 2001, pp. 150–157.

[17] C. Wang, S. Shan, W. Gao, An approach based on phonemes to large vocabulary Chinese sign language recognition, in: Proceedings of International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, 2002, pp. 411–416.

[18] B. Bauer, H. Hienz, Relevant features for video-based continuous sign language recognition, in: Proceedings of International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, 2000, pp. 440–445.

[19] B. Bauer, K.-F. Kraiss, Towards an automatic sign language recognition system using subunits, in: Proceedings of Gesture Workshop, London, UK, 2001, pp. 64–75.

[20] C. Vogler, H. Sun, D. Metaxas, A framework for motion recognition with applications to American sign language and gait recognition, in: Proceedings of Workshop on Human Motion, Austin, TX, 2000, pp. 33–38.

[21] C. Vogler, D. Metaxas, Handshapes and movements: multiple-channel ASL recognition, in: Proceedings of Gesture Workshop, Genova, Italy, 2003, pp. 247–258.

[22] G. Fang, et al., Signer-independent continuous sign language recognition based on SRN/HMM, in: Proceedings of Gesture Workshop, London, UK, 2001, pp. 76–85.

[23] G. Fang, W. Gao, A SRN/HMM system for signer-independent continuous sign language recognition, in: Proceedings of International Conference on Automatic Face and Gesture Recognition, Washington, DC, 2002, pp. 312–317.

[24] A. Farhadi, D. Forsyth, R. White, Transfer learning in sign language, in: Proceedings of Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007, pp. 1–8.

[25] S.C.W. Ong, S. Ranganath, Deciphering gestures with layered meanings and signer adaptation, in: Proceedings of International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, 2004, pp. 559–564.

[26] U. von Agris, D. Schneider, J. Zieren, K.-F. Kraiss, Rapid signer adaptation for isolated sign language recognition, in: Proceedings of Conference on Computer Vision and Pattern Recognition Workshop, New York, 2006, pp. 159–164.

[27] U. von Agris, K.-F. Kraiss, Towards a video corpus for signer-independent continuous sign language recognition, in: Proceedings of Gesture Workshop, Lisbon, Portugal, 2007.

[28] U. von Agris, C. Blömer, K.-F. Kraiss, Rapid signer adaptation for continuous sign language recognition using a combined approach of eigenvoices, MLLR, and MAP, in: Proceedings of International Conference on Pattern Recognition, Tampa, FL, 2008, pp. 1–4.

[29] R. Kuhn, et al., A rapid speaker adaptation in eigenvoice space, IEEE Transactions on Speech and Audio Processing 8 (6) (2000) 695–707.

[30] W.W. Kong, S. Ranganath, Automatic hand trajectory segmentation and phoneme transcription for sign language, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition, Amsterdam, The Netherlands, 2008, pp. 1–6.

[31] S. Sarawagi, W.W. Cohen, Semi-Markov conditional random fields for information extraction, in: Advances in Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2004, pp. 1185–1192.

[32] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, 1st ed., John Wiley and Sons, 1973.

[33] W.W. Kong, S. Ranganath, Sign language phoneme transcription with rule-based hand trajectory segmentation, Signal Processing Systems 59 (2) (2010) 211–222.

[34] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, The MIT Press, 2000, pp. 61–74.

[35] R. Klinger, K. Tomanek, Classical Probabilistic Models and Conditional Random Fields, Algorithm Engineering Report TR07-2-013, Department of Computer Science, Dortmund University of Technology, 2007.

[36] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of International Conference on Machine Learning, 2001, pp. 282–289.

[37] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 972–976.

[38] Virtual Technologies, Inc., CyberGlove® Reference Manual, August 1998.

[39] Polhemus, Inc., 3SPACE® FASTRAK® USER'S MANUAL, rev. c Edition, November 2002.

[40] S.C.W. Ong, Beyond Lexical Meaning: Probabilistic Models for Sign Language Recognition, Ph.D. Thesis, National University of Singapore, 2007.

W.W. Kong received the B.Eng. (Honors) degree in Electrical Engineering (2000), the M.Eng. degree (2005), and the Ph.D degree (2012) both in Electrical and Computer Engineering, from the National University of Singapore. During 2001, she was with the R&D Division of Singapore Epson Industrial Pvt. Ltd., where she was working on scanner software development and testing. Her research interests are in human gesture understanding applications and also include human–computer interaction, machine learning, and computer vision.

Surendra Ranganath received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology Kanpur, the M.E. degree in Electrical Communication Engineering from the Indian Institute of Science Bangalore and the Ph.D. degree in Electrical Engineering from the University of California at Davis. From 1982 to 1985, he was with the Applied Research Group at Tektronix, Inc., Beaverton, OR, where he was working in the area of digital video processing for enhanced and high definition TV. From 1986 to 1991, he was with the medical imaging group at Philips Laboratories, Briarcliff Manor, NY. From 1991 to 2009, he was with the Department of Electrical and Computer Engineering at the National University of Singapore. From 2010 to 2012 he was with the Indian Institute of Technology Gandhinagar. He is currently a Professor in the Information Science and Engineering Department at Sri Jayachamarajendra College of Engineering, Mysore. His research interests are in digital signal and image processing, computer vision, and machine learning with focus on human–computer interaction and video understanding applications.