Fingerspelling recognition with semi-Markov conditional random fields

Taehwan Kim Greg Shakhnarovich Karen Livescu

Toyota Technological Institute at Chicago
6045 S Kenwood Ave, Chicago IL 60637
taehwan, greg, klivescu@ttic.edu

Abstract

Recognition of gesture sequences is in general a very difficult problem, but in certain domains the difficulty may be mitigated by exploiting the domain's "grammar". One such grammatically constrained gesture sequence domain is sign language. In this paper we investigate the case of fingerspelling recognition, which can be very challenging due to the quick, small motions of the fingers. Most prior work on this task has assumed a closed vocabulary of fingerspelled words; here we study the more natural open-vocabulary case, where the only domain knowledge is the possible fingerspelled letters and statistics of their sequences. We develop a semi-Markov conditional model approach, where feature functions are defined over segments of video and their corresponding letter labels. We use classifiers of letters and linguistic handshape features, along with expected motion profiles, to define segmental feature functions. This approach improves letter error rate (Levenshtein distance between hypothesized and correct letter sequences) from 16.3% using a hidden Markov model baseline to 11.6% using the proposed semi-Markov model.

1. Introduction

Recognition of gesture sequences is in general very challenging. However, in some cases there may be a domain "grammar" that can be used to provide a prior on possible gestures and sequences, as in certain forms of dancing, sports, and aircraft marshalling. One of the most practically important of such grammatically constrained gesture sequence domains is sign language.

In this paper we consider American Sign Language (ASL), and focus in particular on recognition of fingerspelled letter sequences. In fingerspelling, signers spell out a word as a sequence of handshapes or hand trajectories corresponding to individual letters. The handshapes used in fingerspelling are also used throughout ASL. In fact, the fingerspelling handshapes account for about 72% of ASL handshapes [7], making research on fingerspelling applicable to ASL in general.

Figure 1 shows the ASL fingerspelling alphabet. Fingerspelling is a constrained but important part of ASL, accounting for up to 35% of ASL [22]. Fingerspelling is typically used for names, borrowings from English or other spoken languages, or new coinages. ASL fingerspelling uses a single hand and involves relatively small and quick motions of the hand and fingers, as opposed to the typically larger arm motions involved in other signs. Therefore, fingerspelling can be difficult to analyze with standard approaches for pose estimation and tracking from video.

Most prior work on fingerspelling recognition has assumed a closed vocabulary of fingerspelled words, often limited to 20-100 words, typically using hidden Markov models (HMMs) representing letters or letter-to-letter transitions [14, 20, 26]. In such settings it is common to obtain letter error rates (Levenshtein distances between hypothesized and true letter sequences, as a proportion of the number of true letters) of 10% or less. In contrast, we address the problem of recognizing *unconstrained* fingerspelling sequences. This is a more natural setting, since fingerspelling is often used for names and other "new" terms, which may not appear in any closed vocabulary.

We develop a semi-Markov conditional random field (SCRF) approach to the unconstrained fingerspelling recognition problem. In SCRFs [28, 41], feature functions are defined over segments of observed variables (in our case, any number of consecutive video frames) and their corresponding labels (in our case, letters). The use of such segmental feature functions is useful for gesture modeling, where it is natural to consider the trajectory of some measurement or the statistics of an entire segment. In this work we define feature functions based on scores of letter classifiers, as well as classifiers of *handshape features* suggested by linguistics research on ASL [6, 19]. Linguistic handshape features summarize certain important aspects of a given letter, such as the "active" fingers or the flexed/non-flexed status

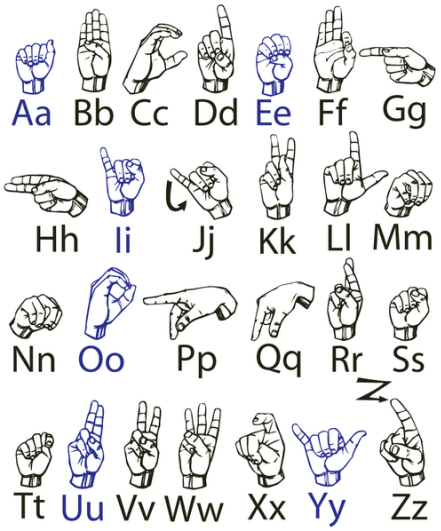

Figure 1. The ASL fingerspelled alphabet. All of the letters except for J and Z are static. [From Wikipedia]

of the fingers; these are hopefully the most salient aspects of each letter, which should be more discriminative for recognition. We also use the statistics of raw visual descriptor derivatives over entire segments to define basic "expected motion" feature functions.

In the remaining sections we describe the approach, its relationship to earlier work, and experimental results demonstrating that our approach provides an improvement over HMM baselines similar to those of prior work.

2. Related work

There has been significant work on sign language recognition from video[1], but there are still large gaps, especially for continuous, large-vocabulary signing settings. Much prior work has involved hidden Markov model (HMM)-based approaches [30, 35, 15, 11], which serves as a natural baseline for our work. There have also been a number of successful approaches using conditional models [40] and more complex (non-linear-chain) graphical models [32, 38].

Some prior work has used representations of handshape and motion that are linguistically motivated, e.g., [5, 10, 32, 38, 37, 36, 35, 23, 33]. However, a much finer level of detail is needed for the sub-articulators of the hand, which motivates our use of linguistic handshape features here.

A subset of ASL recognition work has focused on fingerspelling and/or handshape classification [4, 27, 24] and fingerspelling sequence recognition [14, 20, 26], where letter error rates of 10% or less have been achieved when the recognition is constrained to a small (up to 100-word) lexicon of possible sequences. The only unrestricted fingerspelling recognition work of which we are aware is [19], using HMM-based approaches; we consider this work as the most competitive baseline and compare to it in the experiments section.

The relatively little work on applying segmental (semi-Markov) models to vision tasks has focused on classification and segmentation of action sequences [29, 12] with a small set of possible activities to choose from, including recent work on spotting of specific signs in sign language video [8]. In natural language processing, semi-Markov CRFs have been used for named entity recognition [28], where the labeling is binary. Such models have been applied more widely in speech recognition [41]. One aspect that our work shares with the speech recognition work is that we have a relatively large set of labels (26 letters plus non-letter "N/A" labels), and widely varying lengths of segments corresponding to each label (typically 7-10 video frames, but all lengths from 2-40 frames are seen in our data), which makes the search space large and the inference task correspondingly difficult. We address this difficulty similarly to [41], by adopting a two-stage approach of generating a graph of candidate segmentations and reranking them using the semi-Markov model.

The work presented in this paper is the largest-scale use of semi-Markov models in computer vision, as well as the least constrained fingerspelling recognition experiments, of which we are aware.

3. A semi-Markov CRF approach

We begin by defining the problem and our notation. Let the sequence of visual observations for a given video (corresponding to a single word) be $O = o_1, \ldots, o_T$, where each o_t is a multidimensional image descriptor for frame t. Our goal is to predict the label (letter) sequence. Ideally we would like to predict the best label sequence, marginalizing out different possible label start and end times, but in practice we use the typical approach of predicting the best sequence of frame labels $S = s_1, \ldots, s_T$. We predict S by maximizing its conditional probability under our model, $\hat{S} = \text{argmax}_S P(S|O)$. In generative models like HMMs, we have a joint model $P(S, O)$ and we make a prediction using Bayes' rule. In conditional models we directly represent the conditional distribution $P(S|O)$. For example, in a typical linear-chain CRF, we have:

$$p(S|O) = \frac{1}{Z(O)} \exp\left(\sum_{v,k} \lambda_k f_k(S_v, O_v) + \sum_{e,k} \mu_k g_k(S_e)\right)$$

where $Z(O)$ is the partition function, f_k are the "node" feature functions that typically correspond to the state in a single frame S_v and its corresponding observation O_v, g_k are

[1] A good deal of prior work on sign language recognition has also used other instrumentation, such as specialized gloves or depth maps [17, 16, 21, 39, 13]. These interfaces are sometimes feasible, but video remains more practical in many settings, and we restrict our discussion to video here.

"edge" feature functions corresponding to inter-state edges, e ranges over pairs of frames and S_e is the pair of states corresponding to e, and λ_k and μ_k are the weights.

It may be more natural to consider feature functions that span entire segments corresponding to the same label. Semi-Markov CRFs [28], also referred to as segmental CRFs [41] or SCRFs, provide this ability.

Figure 2 illustrates the SCRF notation, which we now describe. In a SCRF, we consider the segmentation to be a latent variable and sum over all possible segmentations of the observations corresponding to a given label sequence to get the conditional probability of the label sequence $S = s_1, \ldots, s_L$, where the length of S is now the (unknown) number of distinct labels L:

$$p(S|O) = \frac{\sum_{q \text{ s.t. } |q|=|S|} \exp\left(\sum_{e \in q,k} \lambda_k f_k(s_l^e, s_r^e, O_e)\right)}{\sum_{S'} \sum_{q \text{ s.t.} |q|=|S'|} \exp\left(\sum_{e \in q,k} \lambda_k f_k(s_l^e, s_r^e, O_e)\right)}$$

Here, S' ranges over all possible state (label) sequences, q is a segmentation of the observation sequence whose length (number of segments) must be the same as the number of states in S (or S'), e ranges over all state pairs in S, s_l^e is the state which is on the left of an edge, s_r^e is the state on the right of an edge, and O_e is the multi-frame observation segment associated with s_r^e. In our work, we use a baseline frame-based recognizer to generate a set of candidate segmentations of O, and sum only over those candidate segmentations. In principle the inference over all possible segmentations can be done, but typically this is only feasible for much smaller search spaces than ours.

3.1. Feature functions

We define several types of feature functions, some of which are quite general to sequence recognition tasks and some of which are tailored to fingerspelling recognition:

3.1.1 Language model feature

The language model feature is a smoothed bigram probability of the letter pair corresponding to an edge:

$$f_{lm}(s_l^e, s_r^e, O_e) = p_{LM}(s_l^e, s_r^e).$$

3.1.2 Baseline consistency feature

To take advantage of the existence of a high-quality baseline, we use a baseline feature like the one introduced by [41]. This feature is constructed using the 1-best output hypothesis from an HMM-based baseline recognizer. The feature value is 1 when a segment spans exactly one letter label hypothesized by the baseline and the label matches it:

$$f_b(s_l^e, s_r^e, O_e) = \begin{cases} +1 & \text{if } C(t(e), T(e)) = 1, \\ & \text{and } B(t(e), T(e)) = w(s_r^e) \\ -1 & \text{otherwise} \end{cases}$$

where $t(e)$ and $T(e)$ are the start and end times corresponding to edge e, $C(t, T)$ is the number of distinct baseline labels in the time span from t to T, $B(t, T)$ is the label corresponding to time span (t, T) when $C(t, T) = 1$, and $w(s)$ is the letter label of state s.

3.1.3 Handshape classifier-based feature functions

The next set of feature functions measure the degree of match between the intended segment label and the appearance of the frames within the segment. For this purpose we use a set of frame classifiers, each of which classifies either letters or linguistic handshape features. As in [19], we use the linguistic handshape feature set developed by Brentari [6], who proposed seven features to describe handshape in ASL. Each such linguistic feature (not to be confused with feature functions) has 2-7 possible values. Of these, we use the six that are contrastive in fingerspelling. See Table 1 for the details. For each linguistic feature or letter, we train a classifier that produces a score for each feature value for each video frame. We also train a separate letter classifier. We use neural network (NN) classifiers, and consider several types of NN output-layer functions as the classifier scores:

NN output layer types

- linear: $g(v|x) = w_v^{\mathrm{T}} \phi(x)$
- softmax: $g(v|x) = \frac{\exp(w_v^{\mathrm{T}} \phi(x))}{\sum_i \exp(w_i^{\mathrm{T}} \phi(x))}$
- sigmoid: $g(v|x) = \frac{1}{1+\exp(-w_v^{\mathrm{T}} \phi(x))}$

where x is the input to the NN (in our case, the visual descriptors concatenated over a window around the current frame), w_v is the weight vector in the last layer of the NN corresponding to linguistic feature v, and $\phi(x)$ is the input to the last layer. We then use these score functions $g(v|x)$ to define four types of segment feature functions:

Feature functions Let y be a letter and v be the value of a linguistic feature or letter, $N_e = |O_e| = T(e) + 1 - t(e)$ the length of an observation segment corresponding to edge e, and $g(v|o_i)$ the output of a NN classifier at frame i corresponding to class v. We define

- mean: $f_{yv}(s_l^e, s_r^e, O_e) = \delta(w(s_r^e) = y) \cdot \frac{1}{N_e} \sum_{i=t(e)}^{T(e)} g(v|o_i)$
- max: $f_{yv}(s_l^e, s_r^e, O_e) = \delta(w(s_r^e) = y) \cdot \max_{i \in (t(e), T(e))} g(v|o_i)$
- div_s: a concatenation of three mean feature functions, each computed over a third of the segment

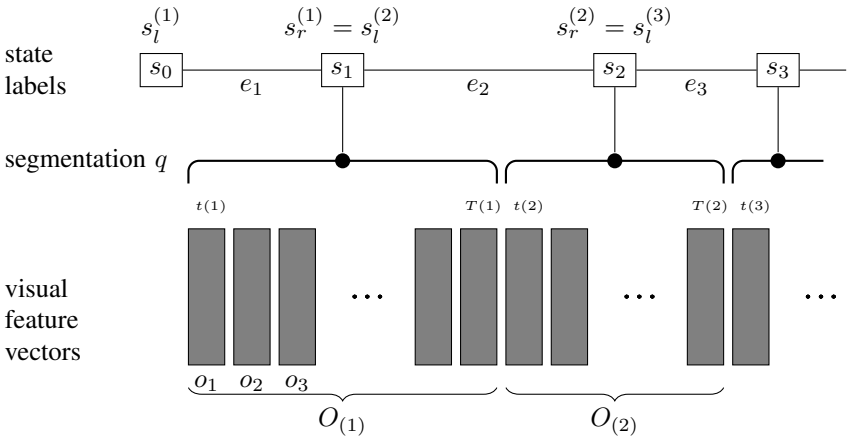

Figure 2. Illustration of SCRF notation. For example, edge e_2 is associated with the "left state" $s_l^{(2)}$, the "right state" $s_r^{(2)}$, and the segment of observations $O_{(2)}$ spanning frames $t(2)$ through $T(2)$.

- div_m: a concatenation of three max feature functions, each computed over a third of the segment

3.1.4 Peak detection features

Fingerspelling a sequence of letters yields a corresponding sequence of "peaks" of articulation. Intuitively, these are frames in which the hand reaches the target handshape for a particular letter. The peak frame and the frames around it for each letter tend to be characterized by very little motion as the transition to the current letter has ended while the transition to the next letter has not yet begun, whereas the transitional frames between letter peaks have more motion. To use this information and encourage each predicted letter segment to have a single peak, we define letter-specific "peak detection features" as follows. We first compute approximate derivatives of the visual descriptors, consisting of the l_2 norm of the difference between descriptors in every pair of consecutive frames, smoothed by averaging over 5-frame windows. We expect there to be a single local minimum in this approximate derivative function over the span of the segment. Then we define the feature function corresponding to each letter y as

$$f_y^{\text{peak}}(s_l^e, s_r^e, O_e) = \delta(w(s_r^e) = y) \cdot \delta_{\text{peak}}(O_e)$$

where $\delta_{\text{peak}}(O_e)$ is 1 if there is only one local minimum in the segment O_e and 0 otherwise.

Feature	Definition/Values
SF point of reference (POR)	side of the hand where SFs are located
	SIL, radial, ulnar, radial/ulnar
SF joints	degree of flexion or extension of SFs
	SIL, flexed:base, flexed:nonbase, flexed:base & nonbase, stacked, crossed, spread
SF quantity	combinations of SFs
	N/A, all, one, one > all, all > one
SF thumb	thumb position
	N/A, unopposed, opposed
SF handpart	internal parts of the hand
	SIL, base, palm, ulnar
UF	open/closed
	SIL, open, closed

Table 1. Definition and possible values for phonological features based on [6]. The first five features are properties of the active fingers (*selected fingers*, SF); the last feature is the state of the inactive or *unselected fingers* (UF). In addition to Brentari's feature values, we add a SIL ("silence") value to the features that do not have an N/A value. For detailed descriptions, see [6].

4. Experiments

Data and annotation We report on experiments using video recordings of four native ASL signers. The data were recorded at 60 frames per second in a studio environment. Each signer signed a list of 300 words as they appeared on a computer screen in front of the signer. There were two non-overlapping lists of 300 words (one for signers 1 and 2, the other for signers 3 and 4). Each word was spelled twice, yielding 600 word instances signed by each signer. The lists contained English and foreign words, including proper names and common English nouns. For comparison with prior work, we use the same data from signers 1 and 2 as [18, 19], as well as additional data from signers 3 and 4. The recording settings, including differences in environ-

ment and camera placement across recording sessions, are illustrated in Figure 3.

Figure 3. Example video frames from the four signers.

The signers indicated the start and end of each word by pressing a button, allowing automatic partition of the recording into a separate video for every word. Every video was verified and manually labeled by multiple annotators with the times and letter identities of the peaks of articulation (see Sec. 3.1.4). The peak annotations are used for the training portion of the data in each experiment to segment a word into letters (the boundary between consecutive letters is defined as the midpoint between their peaks).

Hand localization and segmentation For every signer, we trained a model for hand detection similar to that used in [19, 20]. Using manually annotated hand regions, marked as polygonal regions of interest (ROI) in 30 frames, we fit a mixture of Gaussians P_{hand} to the color of the hand pixels in L*a*b color space. Using the same 30 frames, we also built a single-Gaussian color model P_{bg}^{x} for every pixel x in the image excluding pixel values in or near marked hand ROIs. Then, given a test frame, we label each pixel as hand or background based on an odds ratio: Given the color triplet $\mathbf{c}_x = [l_x, a_x, b_x]$ at pixel x, we assign it to hand if

$$P_{hand}(\mathbf{c}_x)\pi_{hand} > P_{bg}^{x}(\mathbf{c}_x)(1-\pi_{hand}), \qquad (1)$$

where the prior π_{hand} for hand size is estimated from the same 30 training frames.

Since this simple model produces rather noisy output, we next clean it up by a sequence of filtering steps. We suppress pixels that fall within regions detected as faces by the Viola-Jones face detector [34], since these tend to be false positives. We also suppress pixels that passed the log-odds test but have a low estimated value of P_{hand}. These tend to correspond to movements in the scene, e.g., a signer changing position and thus revealing previously occluded portions of the background; for such pixels the value of P_{bg} may be low, but so is P_{hand}. Finally, we suppress pixels outside of a (generous) spatial region where the signing is expected to occur. The largest surviving connected component of the resulting binary map is treated as a mask that defines the detected hand region. Some examples of resulting hand regions are shown in Figures 6 and 7. Note that while this procedure currently requires manual annotation for a small number of frames in our offline recognition setting, it could be fully automated in a realistic interactive setting, by asking the subject to place his/her hand in a few defined locations for calibration.

Handshape descriptors The visual descriptor for a given hand region is obtained by concatenation of histograms of oriented gradients (HOG [9]) descriptors, computed on a spatial pyramid of regions over the tight bounding box of the hand region, resized to canonical size of 128×128 pixels. Pixels outside of the hand mask are ignored in this computation. The HOG pyramid consists of 4×4, 8×8, and 16×16 grids, with eight orientation bins per grid cell, resulting in 2688-dimensional descriptors. To speed up computation, these descriptors were projected to at most 200 principal dimensions; the exact dimensionality in each experiment was tuned on a development set (see below).

Letter and linguistic feature classifiers We use feed-forward neural network classifiers (NNs) (trained with Quicknet [25]) for letters and linguistic feature labels for each video frame. The inputs to the NNs are the HOG descriptors concatenated over a window of several frames around each frame. The training labels are obtained from the manually labeled peak frames, as described above. We tune the window sizes on the development set.

Baselines We compare our approach with two HMM baselines, which reproduce the work in [19] except for differences in our hand segmentation and visual descriptors. Both baselines have one 3-state HMM per letter, plus a separate HMM for the sequence-initial and sequence-final non-signing portions (referred to as "n/a"), Gaussian mixture observation densities, and a letter bigram language model. The basic HMM-based recognizer uses dimensionality-reduced visual descriptors directly as observations. The second baseline uses as observations the linear outputs of the NN linguistic feature classifiers, reproducing the "tandem" approach of [19]. This was done to confirm that we can reproduce the result of [19] showing an advantage for the tandem system over standard HMMs. Although we reproduce the models of [19] for comparison, we tune all hyperparameters of each model on held-out data.

Generating candidate segmentations for SCRFs As described above, we use a two-phase inference approach where a baseline recognizer produces a set of candidate segmentations and label sequences, and a SCRF is used to re-rank the baseline candidates. We produce a list of

N-best candidate segmentations using the tandem baseline (as it is the better performer of the two baselines). Training of the SCRFs is done by maximizing the conditional log likelihood under the model, using the Segmental CRF (SCARF) toolkit [1]. For those training examples where the correct letter sequence is not among the baseline N-best candidates, we have several choices. We can add a correct candidate by using the baseline recognizer to align the video to the correct labels (by performing a Viterbi search constrained to state sequences corresponding to the correct letters); use ground truth annotation as a correct candidate segmentation; choose the best possible matching segmentation from the N-best candidates; or finally, discard such examples from training data. In each experiment, we chose among these options by tuning on the development set.

Experimental setup For each signer, we use a 10-fold setup: In each fold, 80% of the data is used as a training set, 10% as a development set for tuning hyperparameters, and the remaining 10% as a final test set. We report the average results over the 10 test sets. We independently tune the parameters in each fold (that is, we run 10 separate, complete experiments) and report the average letter error rate (LER) over the 10 folds. We train the letter bigram language models from large online dictionaries of varying sizes that include both English words and names [2]. We use HTK [3] to implement the baseline HMM-based recognizers and SRILM [31] to train the language models.

The HMM parameters (number of Gaussians per state, size of language model vocabulary, transition penalty and language model weight), as well as the dimensionality of the HOG descriptor input and HOG depth, were tuned to minimize development set letter error rates for the baseline HMM system. For the NN classifiers, the input window size was tuned to minimize frame error rate of the classifiers on the development set. All of the above parameters were kept fixed for the SCRF (in this sense the SCRF is slightly disadvantaged). The NN output type was tuned separately for the tandem HMM and the SCRF. Finally, additional parameters tuned for the SCRF models included the N-best list sizes, type of feature functions, choice of language models, and L1 and L2 regularization parameters.

Results The main results are shown in Figure 4. First, we confirm that the tandem baseline improves over a standard HMM baseline. Second, we find that the proposed SCRF improves over the tandem HMM-based system, correcting about 21% of the errors (or 29% of the errors committed by the basic HMM baseline). For reference, we also show the performance of the NN classifiers in Figure 5. Note that in our experimental setup, there is some overlap of word types between training and test data. This is a realistic setup, since in real applications some of the test words will have been previously seen and some will be new. However, for

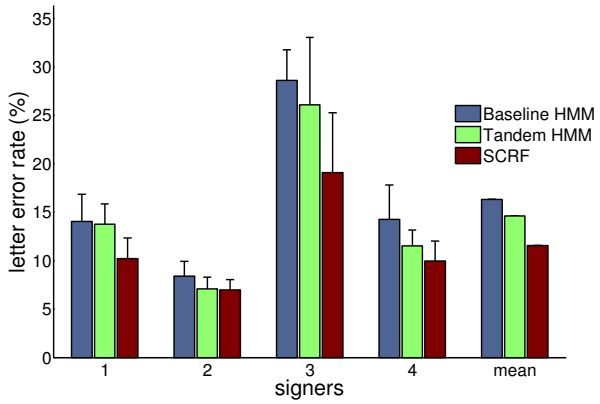

Figure 4. Letter error rate for each signer, and average letter error rate over all signers, for the two baselines and for the proposed SCRF. Error bars show the standard deviations over the 10 folds.

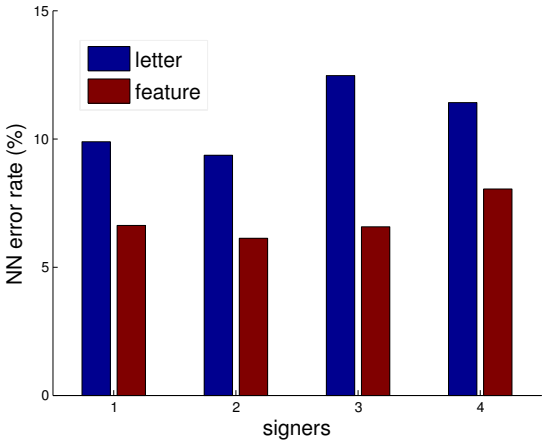

Figure 5. Neural network classifier error rates on letter classification (blue) and linguistic feature classification (red) on test data for each signer. The linguistic feature error rates are averaged over the six linguistic feature classification tasks; error rates for each linguistic feature type range between 4% and 10%. Chance error rates (based on always predicting the most common class) range from approximately 25% to approximately 55%.

comparison, we have also conducted the same experiments while keeping the training, development, and test vocabularies disjoint; in this modified setup, letter error rates increase by about 2-3% overall, but the SCRFs still outperform the other models.

Figures 6 and 7 illustrate the recognition task, showing examples in which the SCRF corrected mistakes made by the tandem HMM recognizer. In each of these figures we show the ground truth segments. The peak frames are shown on top of each letter's segment; the hand region segmentation masks were obtained automatically using the probabilistic model described in Section 4. We also show intermediate frames, obtained at midpoints between peaks,

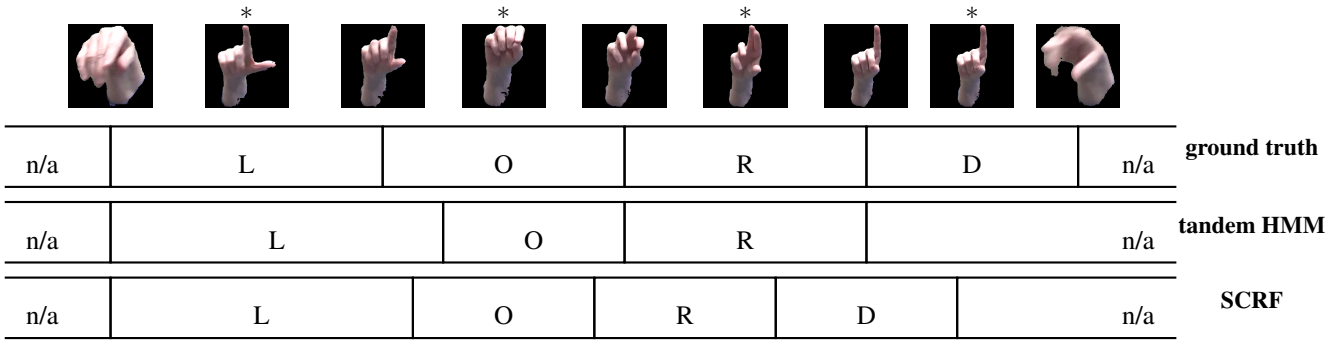

Figure 6. From top to bottom: frames from the word LORD, with asterisks denoting the peak frame for each letter and n/a denoting periods before the first letter and after the last letter; ground-truth segmentation based on peak annotations; segmentation produced by the tandem HMM; and segmentation produced by the proposed SCRF.

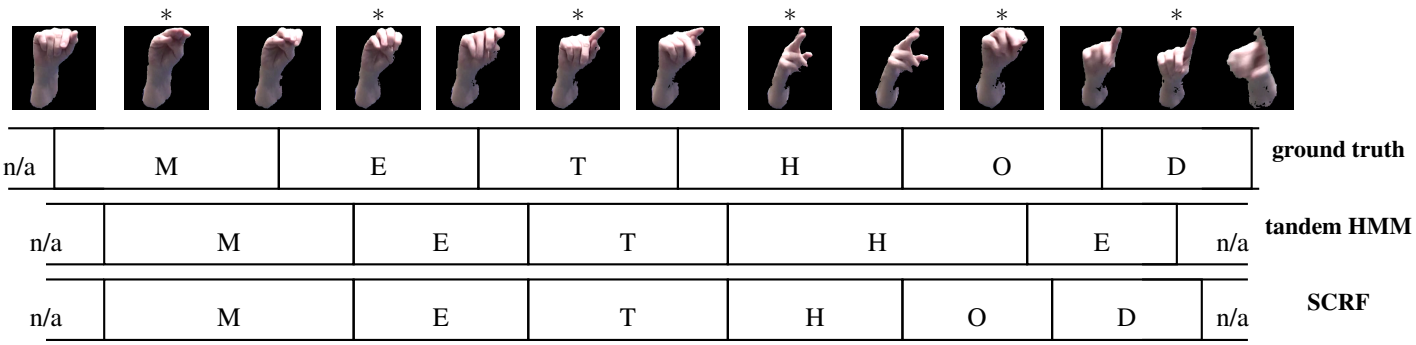

Figure 7. Similar to Figure 6 for the word METHOD.

as well as a frame before the first peak and after the last peak. Below the ground truth segmentations are the segmentations obtained with the baseline tandem HMM, and at the bottom are segmentations obtained with the SCRF.

5. Discussion

This paper proposes an approach to automatic recognition of fingerspelled words in ASL, in a challenging open-vocabulary scenario. This is a compelling task, due to its complexity and its practical importance to a large community of Deaf persons in the US. Our experiments demonstrate the performance of a semi-Markov CRF model on this challenging task. We report results superior to those obtained with the model of [19], which to our knowledge is the only published work that has addressed similar unconstrained recognition settings. The work has implications for the larger task of general ASL recognition, where the same handshapes used in fingerspelling are used throughout.

We believe that our results are promising in a broader context of recognition of action sequences (and in particular gesture sequences) with any sort of "grammar" – constraints that limit a set of configurations, and introduce structure into statistics of possible transitions. Part of our future work is to investigate applications of the proposed SCRF model to other scenarios in which a complex activity of interest can be parsed into a sequence of identifiable building blocks (primitives). Examples of such structured activities include dancing, aircraft marshalling, and others. We also are investigating more complex segmental feature functions that would capture additional properties of the data. Finally, although user-dependent sign language recognition could be useful in practice, as evidenced by the prevalence of such applications for spoken language recognition (such as dictation systems), we would like to develop methods that are more signer-independent.

References

[1] http://research.microsoft.com/en-us/projects/scarf/. 6

[2] http://www.keithv.com/software/csr. 6

[3] http://htk.eng.cam.ac.uk. 6

[4] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. BoostMap: A method for efficient approximate similarity rankings. In *CVPR*, 2004. 2

[5] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *ECCV*, 2004. 2

[6] D. Brentari. *A Prosodic Model of Sign Language Phonology*. MIT Press, 1998. 1, 3, 4

[7] D. Brentari and C. Padden. Native and foreign vocabulary in American Sign Language: A lexicon with multiple origins.

In *Foreign vocabulary in sign languages: A cross-linguistic investigation of word formation*, pages 87–119. Lawrence Erlbaum, Mahwah, NJ, 2001. 1

[8] S.-S. Cho, H.-D. Yang, and S.-W. Lee. Sign language spotting based on semi-Markov conditional random field. In *Workshop on the Applications of Computer Vision*, 2009. 2

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 5

[10] L. Ding and A. M. Martínez. Modelling and recognition of the linguistic components in American Sign Language. *Image Vision Comput.*, 27(12), 2009. 2

[11] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, , and H. Ney. Speech recognition techniques for a sign language recognition system. In *Interspeech*, 2007. 2

[12] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *CVPR*, 2005. 2

[13] R. S. Feris, M. Turk, R. Raskar, K. Tan, and G. Ohashi. Exploiting depth discontinuities for vision-based fingerspelling recognition. In *IEEE Workshop on Real-Time Vision for Human-Computer Interaction*, 2004. 2

[14] P. Goh and E.-J. Holden. Dynamic fingerspelling recognition using geometric and motion features. In *ICIP*, 2006. 1, 2

[15] K. Grobel and M. Assan. Isolated sign language recognition using hidden Markov models. In *International Conference on System Man and Cybernetics*, 1997. 2

[16] K. Grobel and H. Hienz. Video-based recognition of fingerspelling in real-time. In *Workshops Bildverarbeitung für die Medizin*, 1996. 2

[17] M. W. Kadous. Machine recognition of Auslan signs using powergloves: towards large lexicon integration of sign language. In *Workshop on the Integration of Gesture in Language and Speech*, 1996. 2

[18] J. Keane, D. Brentari, and J. Riggle. Coarticulation in ASL fingerspelling. In *NELS*, 2012. 4

[19] T. Kim, K. Livescu, and G. Shakhnarovich. American Sign Language fingerspelling recognition with phonological feature-based tandem models. In *IEEE Workshop on Spoken Language Technology*, 2012. 1, 2, 3, 4, 5, 7

[20] S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in British Sign Language. In *2nd IEEE Workshop on CVPR for Human Communicative Behavior Analysis*, 2009. 1, 2, 5

[21] C. Oz and M. C. Leu. Recognition of finger spelling of American Sign Language with artificial neural network using position/orientation sensors and data glove. In *2nd international conference on Advances in Neural Networks*, 2005. 2

[22] C. Padden and D. C. Gunsauls. How the alphabet came to be used in a sign language. *Sign Language Studies*, page 4:1033, 2004. 1

[23] V. Pitsikalis, S. Theodorakis, C. Vogler, and P. Maragos. Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition. In *IEEE CVPR Workshop on Gesture Recognition*, 2011. 2

[24] N. Pugeault and R. Bowden. Spelling it out: Real-time ASL fingerspelling recognition. In *ICCV*, 2011. 2

[25] http://www.icsi.berkeley.edu/Speech/qn.html. 5

[26] S. Ricco and C. Tomasi. Fingerspelling recognition through classification of letter-to-letter transitions. In *ACCV*, 2009. 1, 2

[27] A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Affine-invariant modeling of shape-appearance images applied on sign language handshape classification. In *ICIP*, 2010. 2

[28] S. Sarawagi and W. W. Cohen. Semi-Markov conditional random fields for information extraction. In *NIPS*, 2004. 1, 2, 3

[29] Q. Shi, L. Cheng, L. Wang, and A. Smola. Human action segmentation and recognition using discriminative semi-Markov models. *International Journal of Computer Vision*, 93(1), 2011. 2

[30] T. Starner, J. Weaver, and A. Pentland. Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12), 1998. 2

[31] A. Stolcke, J. Zheng, W. Wang, and V. Abrash. SRILM at sixteen: update and outlook. In *ASRU*, 2011. 6

[32] A. Thangali, J. P. Nash, S. Sclaroff, and C. Neidle. Exploiting phonological constraints for handshape inference in ASL video. In *CVPR*, 2011. 2

[33] S. Theodorakis, V. Pitsikalis, and P. Maragos. Model-level data-driven sub-units for signs in videos of continuous sign language. In *ICASSP*, 2010. 2

[34] P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 5

[35] C. Vogler and D. Metaxas. Parallel hidden Markov models for American Sign Language recognition. In *ICCV*, 1999. 2

[36] C. Vogler and D. Metaxas. Toward scalability in ASL recognition: Breaking down signs into phonemes. In *Gesture Workshop*, 1999. 2

[37] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of American Sign Language. *Computer Vision and Image Understanding*, 81:358–384, 2001. 2

[38] C. Vogler and D. Metaxas. Handshapes and movements: Multiple-channel ASL recognition. In *Gesture Workshop*, 2003. 2

[39] R. Wang and J. Popovic. Real-time hand-tracking with a color glove. In *SIGGRAPH*, 2009. 2

[40] R. Yang and S. Sarkar. Detecting coarticulation in sign language using conditional random fields. In *ICPR*, 2006. 2

[41] G. Zweig and P. Nguyen. Segmental CRF approach to large vocabulary continuous speech recognition. In *ASRU*, 2009. 1, 2, 3