AMERICAN SIGN LANGUAGE FINGERSPELLING RECOGNITION WITH PHONOLOGICAL FEATURE-BASED TANDEM MODELS

Taehwan Kim, Karen Livescu, Gregory Shakhnarovich

Toyota Technological Institute at Chicago, Chicago, IL, USA

{taehwan,klivescu,greg}@ttic.edu

ABSTRACT

We study the recognition of fingerspelling sequences in American Sign Language from video using tandem-style models, in which the outputs of multilayer perceptron (MLP) classifiers are used as observations in a hidden Markov model (HMM)-based recognizer. We compare a baseline HMM-based recognizer, a tandem recognizer using MLP letter classifiers, and a tandem recognizer using MLP classifiers of phonological features. We present experiments on a database of fingerspelling videos. We find that the tandem approaches outperform an HMM-based baseline, and that phonological feature-based tandem models outperform letter-based tandem models.

Index Terms— American Sign Language, fingerspelling, tandem models, phonological features

1. INTRODUCTION

Automatic sign language recognition has close connections with both speech recognition and computer vision. Progress in sign language recognition has the potential to improve the ability of deaf individuals to communicate with computer systems via untethered interfaces, as well as the ability of deaf and hearing individuals to communicate with each other. The linguistics of sign languages is less well understood than that of spoken languages, and sign language processing technologies are much less advanced than speech technologies. Nevertheless, a significant amount of research effort has been devoted to the problem, including work on visual features of appearance and motion and on statistical models of sign [1, 2, 3, 4, 5, 6].

We consider American sign language (ASL), and focus on recognition of one constrained but important part of the language: fingerspelling, in which signers spell out a word as a sequence of handshapes or hand trajectories corresponding to individual letters. Figure 1 shows the ASL fingerspelling alphabet, and Figure 3 shows images of several example letter signs. Fingerspelling accounts for 12-35% of ASL [7] and is typically used for names and borrowings from English. It differs from other components of ASL in that it involves the motion of a single hand. Fingerspelling involves relatively small and fast motions of the hand and fingers, as opposed to the typically larger arm motions involved in other signs. Therefore, fingerspelling is difficult to analyze with standard algorithms for pose estimation and tracking from video.

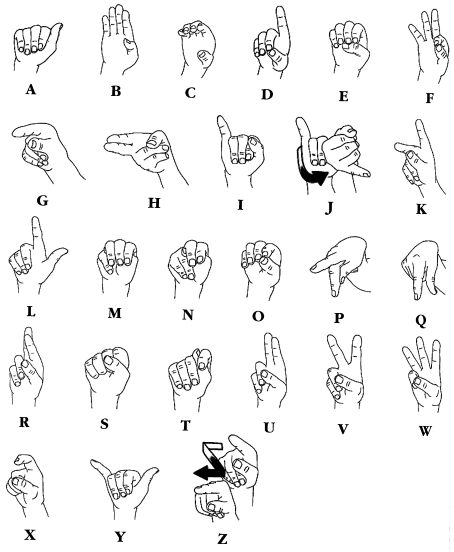

Fig. 1. The ASL fingerspelled alphabet. Reproduced from [8].

We focus on fingerspelling for several reasons. First, research on sign language recognition typically focuses on the larger motions involved in signing, and relatively little attention has been placed on the aspects of handshape. Fingerspelling allows us to focus exclusively on handshape. Second, while analysis of fingerspelling motions is quite challenging for standard computer vision algorithms, using linguistic constraints and ideas from spoken language technology can help to constrain the task and allow for more successful recognition. The field of linguistics has been developing rich models of the phonology of handshape [9, 10], and in this work we use some of those ideas.

Finally, despite the widespread use of fingerspelling in ASL, there has been relatively little prior research on fingerspelling recognition. The problem can be approached similarly to speech recognition, with letters being the ana-

logues of words or phones, and most work thus far has indeed used HMM-based approaches with HMMs representing letters (e.g., [11, 4]) or letter-to-letter transitions [12]. Most prior work has limited the vocabulary to 20-100 words, in which case it is common to obtain letter error rates of 10% or less. In this work, we are interested in recognition of arbitrary fingerspelling sequences with an unconstrained vocabulary. This is a natural setting, since fingerspelling is often used for names and other "out-of-vocabulary" terms.

As in most previous work, we begin with an HMM baseline, using image appearance features as observations. We then consider the tandem approach to speech recognition [13], in which the outputs of multilayer perceptron (MLP) classifiers of phones are used as observations in HMM-based recognition. We first adapt this approach, using MLP classifiers of individual fingerspelled letters. Next, we propose a tandem approach using MLP classifiers of phonological features of fingerspelling [9] rather than of letters. Unlike prior work on fingerspelling, we study the case where there is no known vocabulary of words that limits the possible letter sequences.

2. A TANDEM APPROACH

We base our approach to fingerspelling recognition on the popular tandem approach to speech recognition [13]. In tandem-based speech recognition, multilayer perceptrons (MLPs) are trained to classify phones, and their outputs (phone posteriors) are post-processed and used as observations in a standard HMM-based recognizer with Gaussian mixture observation distributions. The post-processing may include taking the logs of the posteriors (or simply taking the linear outputs of the MLPs rather than posteriors), applying principal components analysis, and/or appending acoustic features to the MLP outputs.

In this work, we begin with a basic adaptation of the tandem approach, where instead of phone posteriors estimated from acoustic frames, we use letter posteriors estimated from image features (described in Section 3). Next, we propose a tandem model using classifiers of phonological features of fingerspelling rather than of the letters themselves. The motivation is that, since features have fewer values, it may be possible to learn them more robustly than letters from small training sets, and certain features may be more or less difficult to classify. This is similar to the work in speech recognition of Çetin et al. [14], who used articulatory feature MLP classifiers rather than phone classifiers.

We use a phonological feature set developed by Brentari [9], who proposed seven features for ASL handshape. Of these, we use the six that are contrastive in fingerspelling. The features and their values are given in Table 1. Example frames for values of the "SF thumb" feature are shown in Figure 2, and entire phonological feature vectors for several letters are shown in Figure 3. We also show examples for one of those features, thumb, in Figure 2. The

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. Definitions and values of phonological features based on [9]. The first five features refer to the active fingers (*selected fingers*, SF); the last is the state of the inactive or *unselected* fingers (UF). In addition to Brentari's feature values, we add a SIL ("silence") value to features that lack an N/A value. For more details, see [9].

observations used in our HMMs consist of functions of the MLP posteriors for all values of all MLPs, concatenated with image appearance features. In the feature-based tandem case, there is a total of 26 posteriors per frame; in the letter-based tandem case, there are 28 (26 letters + beginning "silence" + ending "silence").

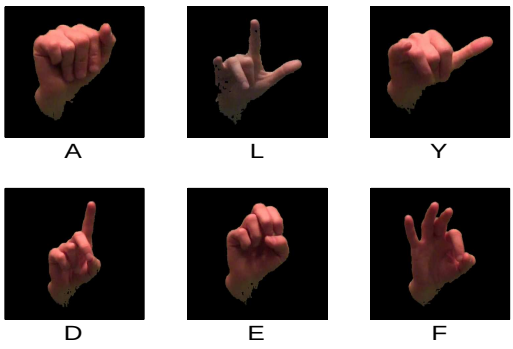

Fig. 2. Example images corresponding to SF thumb = 'unopposed' (upper row) and SF thumb = 'opposed' (bottom row).

Fingerspelling-specific challenges. In many ways the problem of fingerspelling recognition is analogous to that of word sequence or phone sequence recognition. However, there are some special challenges that fingerspelling introduces. We

address some of these challenges here, while others are left to future work.

First, the concept of a "silence" unit is quite different from that of acoustic silence. By "silence", we mean any video segment that does not correspond to fingerspelling, including any hand motion that is not linguistically meaningful. This may be thought of as a "garbage" unit, but its appearance is highly dependent on the context. In our data, for example, "silence" typically corresponds to the time at the beginning and end of each letter sequence, when the signer's hand is rising to/falling from the signing position. We handle "silences" by having separate HMMs for utterance-initial and utterance-final silence, as well as inter-letter "short pause" to cover additional motions such as hesitation. The utterance-initial and utterance-final silence models have several distinct states, to account for the consistent dynamics of these units.

Second, double letters are usually not signed as two copies of the same letter, but rather as either a single longer articulation or a special sign for the doubled letter. For example, 'ZZ' is often signed identically to a 'Z' but using two extended fingers rather than the usual one. We have attempted several solutions to this problem, such as having distinct HMMs for the different realizations, but this requires more data than we currently have available. For the work presented here, we treat double letters produced with a single articulation as a single letter, and ignore special double-letter signs.

Finally, the language model over fingerspelled letter sequences is difficult to estimate. There is no large database of natural fingerspelled sequences in running sign. In our data set, the distribution of words has been chosen to maximize coverage of letter n-grams and word types rather than to follow some natural distribution. Fingerspelling does not follow the distribution of, say, English words, since fingerspelling is most often used for words lacking ASL signs, such as names. For this work, we estimate language models from large English dictionaries that include names. However, such language models are not a perfect fit, and this issue requires more attention.

3. EXPERIMENTS

This work uses newly collected data from two adult native ASL signers, recorded in a studio environment [15]. Each signer fingerspelled words from a list of 300 words (100 English nouns, 100 English names, and 100 non-English words) intended to cover a broad variety of letter combinations. Each word was fingerspelled twice, and the signer indicated the start and end of the word by pressing a button. While this data set is small, it is relatively large compared to previous fingerspelling data collection efforts. The data has been manually labeled and verified by multiple annotators, with the times and letter identities of *apogees*, or image frames corresponding to the peak of articulation of each letter. No other labels other than the start and end button presses and apogee frames are available. More information about the data collection can be found in [15]. Examples of images from the data set are shown in Figure 3.

Image processing and feature extraction. The data has been recorded at 60 image frames per second. The hand is automatically segmented from each image using a color-based model (mixture of Gaussians for skin color vs. single Gaussian per background pixel). We then extract SIFT (scale-invariant feature transform) features, a commonly used type of image appearance features, from each image, based on local histograms of oriented image gradients [16]. To capture both local and global information, we concatenate SIFT features computed over a spatial pyramid in the image (the entire image, image quarters, etc.). The resulting feature vectors have a few thousand dimensions, which we reduce using principal components analysis (PCA).

Multilayer perceptrons. The inputs to the MLPs are the SIFT features concatenated over a window of several frames around each frame. The MLPs are implemented with Quicknet [17]. We consider several choices for the MLP output functions: posteriors (softmax), log posteriors, and linear outputs. We obtain MLP training labels for each frame either from the manually labeled apogees or from forced alignments produced by the baseline HMM-based recognizer. To derive a letter label for each frame given only the apogees, we assume that there is a letter boundary in the middle of each segment between consecutive apogees. We use an MLP with one hidden layer with 1000 hidden nodes.

Experimental setup. The task is recognition of one fingerspelled word at a time, delimited by the signer's button presses. We use a speaker-dependent 10-fold setup: We divide each speaker's data ($\sim$ 600 words, $\sim$ 3000 letters) randomly into ten subsets. In each fold, eight of the subsets (80% of the data) are used for training both MLPs and HMMs, one (10%) for tuning MLP and HMM parameters, and one (10%) as a final test set. We implement the HMMs with HTK [18] and language models with SRILM [19]. We train smoothed backoff bigram letter language models using lexicons of various sizes, consisting of the most frequent words in the ARPA CSR-III text, which includes English words and names [20].

The tuning parameters, and their most frequently chosen values, are the SIFT pyramid depth (1+2), PCA dimensionality (80), window size (13), MLP output function (linear), whether or not SIFT features are appended to MLP outputs (yes), number of states per letter HMM (3), number of silence states (9), number of Gaussians per state (8), and language model lexicon size (5k-word). We tune independently in each fold, i.e., we perform ten independent experiments, and report the average test performance over the folds. Recognition performance is measured via the letter error rate, the Levenshtein distance between the hypothesized letter sequence and the reference sequence as a percentage of the reference sequence length.

Figure 3 shows an example of the operation of the recognizers, including most of the above components.

Results. Figure 4 gives the frame error rates of the MLPs, measured with respect to frame labels produced from the manual apogee labels as described above. All of the classifiers perform much better than chance.

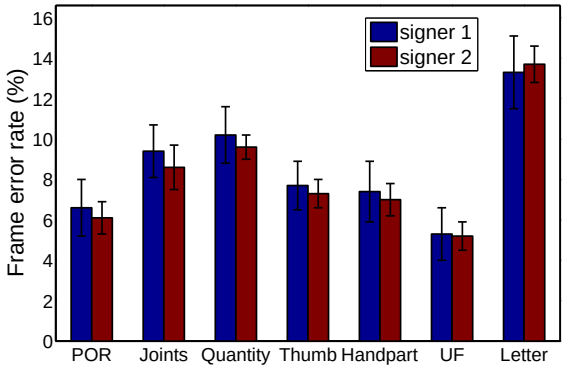

Fig. 4. Frame error rates of letter and feature MLPs, averaged over ten folds, and their standard deviations. Chance performance (100 - frequency of the most likely class) is $\sim 25\%$ to $\sim 55\%$, depending on the classifier.

Figure 5 shows our main results: a comparison of the two tandem models and the baseline HMM-based system, using different types of training labels. For the baseline system, there are two choices for training: either (1) the manual apogee labels are used to generate a segmentation into letters, and each letter HMM is trained only on the corresponding segments; or (2) the manual labels are ignored and the HMMs are trained without any segmentation, using Expectation-Maximization on sequences corresponding to entire words. For the tandem systems, we consider three choices: (1) the manual labels are used to generate a segmentation, and both the MLPs and HMMs are trained using this labeled segmentation; (2) the segmentation is used for MLP training, but HMMs are trained without it; or (3) the segmentation is not used at all, and the labels for MLP training are derived via forced alignment using the baseline (segmentation-free) HMM. The reason for this comparison is to determine to what extent the (rather time-intensive) manual labeling is helpful.

The tandem systems consistently improve over the corresponding baselines, with the phonological feature-based tandem models outperforming the letter-based tandem models. Using the manual labels in training makes a large difference to all of the recognizers' error rates, with the forced alignment-based labels producing poorer performance (iterating the forced alignment procedure does not help). However, in all cases the tandem-based models improve over the baselines and the feature-based models improve over the letter-based tandem models. Reducing our dependence on manual labels is one area for future work.

The most commonly confused letter pairs for our baseline system are (U,R), (Q,G), (Y,X), (O,E), (T,N), (J,I), (X,E), (T,S), and (V,W). Of these, our best feature tandem system recovers about 60% of the (U,R) errors, 40% of the (O,E) errors, and all of the (X,E) and (V,W) errors. In all of these cases, the letters are quite similar, often differing by only one feature. For example, (Q,G) differ in the orientation of the hand, and (U,R) differ in whether the selected fingers are crossed. Figure 6 gives example images of Q misrecognized as G.

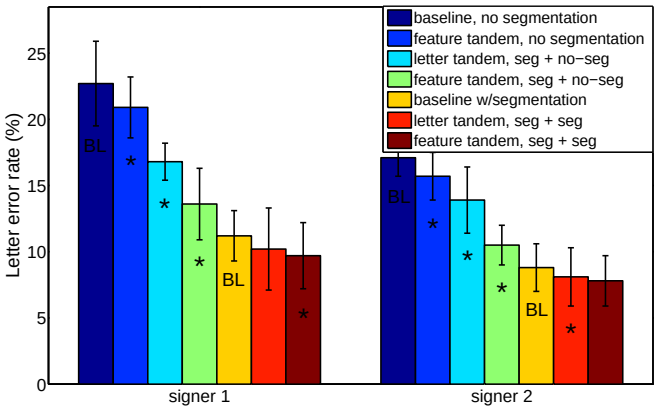

Fig. 5. Letter error rates and standard deviations on two signers. For tandem systems, "seg + no-seg" indicates that segmentation based on manual apogee labels was used for MLP training but not for HMM training; "seg + seg" = the segmentation was used for both MLP and HMM training; "no segmentation" = forced alignment using the HMM baseline was used to generate MLP training labels. An asterisk ('*') indicates statistically significant improvement over the corresponding baseline ('BL') using the same training labels for the HMMs, according to a MAPSSWE test [21] at $p < 0.05$.

Fig. 6. Example images of Q recognized as G.

As a point of reference, if we restrict the output of our best system to the known 300-word vocabulary, the error rates are extremely low—2.6% for Signer 1 and 0.6% for Signer 2—similarly to prior work [4]. However, we are interested in the more challenging case of an unknown vocabulary.

4. CONCLUSION

The main results from this work are that the tandem models outperform an HMM baseline, and that the phonological feature-based tandem models outperform letter-based tandem models. The tandem systems, of course, require per-frame

letter labels in training, although the gains we report are obtained using only labeled apogees and not full manual letter alignments. When we discard the apogee labels, both the baseline and the tandem models perform significantly worse. One area for future work, therefore, is automatic or semi-automatic apogee detection and labeling, as well as semi-supervised learning to minimize the amount of manual labor involved.

The data set used here is in the process of being further annotated and expanded to include more signers, varying signing rates, and more challenging visual conditions. Future work will focus on these more challenging settings, including signer independence and/or adaptation. For the more challenging visual conditions, more sophisticated visual analysis may be needed, for example explicitly accounting for motion, pose, or 3-D structure. From a linguistic perspective, we are interested in the coarticulation that occurs in fingerspelling at different rates [15], and believe that phonological feature models may be particularly well-suited to handle such effects.

5. ACKNOWLEDGMENTS

We thank Diane Brentari, Jonathan Keane, and Jason Riggle for their comments and help with the data and features, and Arild Næss for comments on an earlier draft.

6. REFERENCES

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

[2] M. M. Zaki and S. I. Shaheen, "Sign language recognition using a combination of new vision based features," *Pattern Recognition Letters*, pp. 3397–3415, 2010.

[3] 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.

[4] S. Liwicki and M. Everingham, "Automatic recognition of fingerspelled words in British sign language," in *CVPR*, 2009.

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

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

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

[8] T. Humphries and C. Padden, *Learning American Sign Language (Levels I & II)*, Pearson Education, New York, second edition, 2004.

[9] D. Brentari, *A Prosodic Model of Sign Language Phonology*, MIT Press, 1998.

[10] R. E. Johnson and S. K. Liddell, "Toward a phonetic representation of signs: sequentiality and contrast," *Sign Language Studies*, vol. 11, no. 2, pp. 241–274, 2010.

[11] P. Goh and E.-J. Holden, "Dynamic fingerspelling recognition using geometric and motion features," in *ICIP*, 2006.

[12] S. Ricco and C. Tomasi, "Fingerspelling recognition through classification of letter-to-letter transitions," in *ACCV*, 2009.

[13] D. P. W. Ellis, R. Singh, and S. Sivadas, "Tandem acoustic modeling in large-vocabulary recognition," in *ICASSP*, 2001.

[14] O. Çetin, A. Kantor, S. King, C. Bartels, M. Magimai-Doss, J. Frankel, and K. Livescu, "An articulatory feature-based tandem approach and factored observation modeling," in *ICASSP*, 2007.

[15] J. Keane, D. Brentari, and J. Riggle, "Coarticulation in ASL fingerspelling," in *NELS*, 2012.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[17] D. Johnson et al., "ICSI quicknet software package," `http://www.icsi.berkeley.edu/Speech/qn.html`, 2004.

[18] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*, Cambridge University Engineering Department, 2002, `http://htk.eng.cam.ac.uk`.

[19] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at sixteen: update and outlook," in *ASRU*, 2011.

[20] D. Graff, R. Rosenfeld, and D. Paul, "CSR-III text," `http://http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC95T6`, 1995.

[21] D. S. Pallet, W. M. Fisher, and J. G. Fiscus, "Tools for the analysis of benchmark speech recognition tests," in *ICASSP*, 1990.

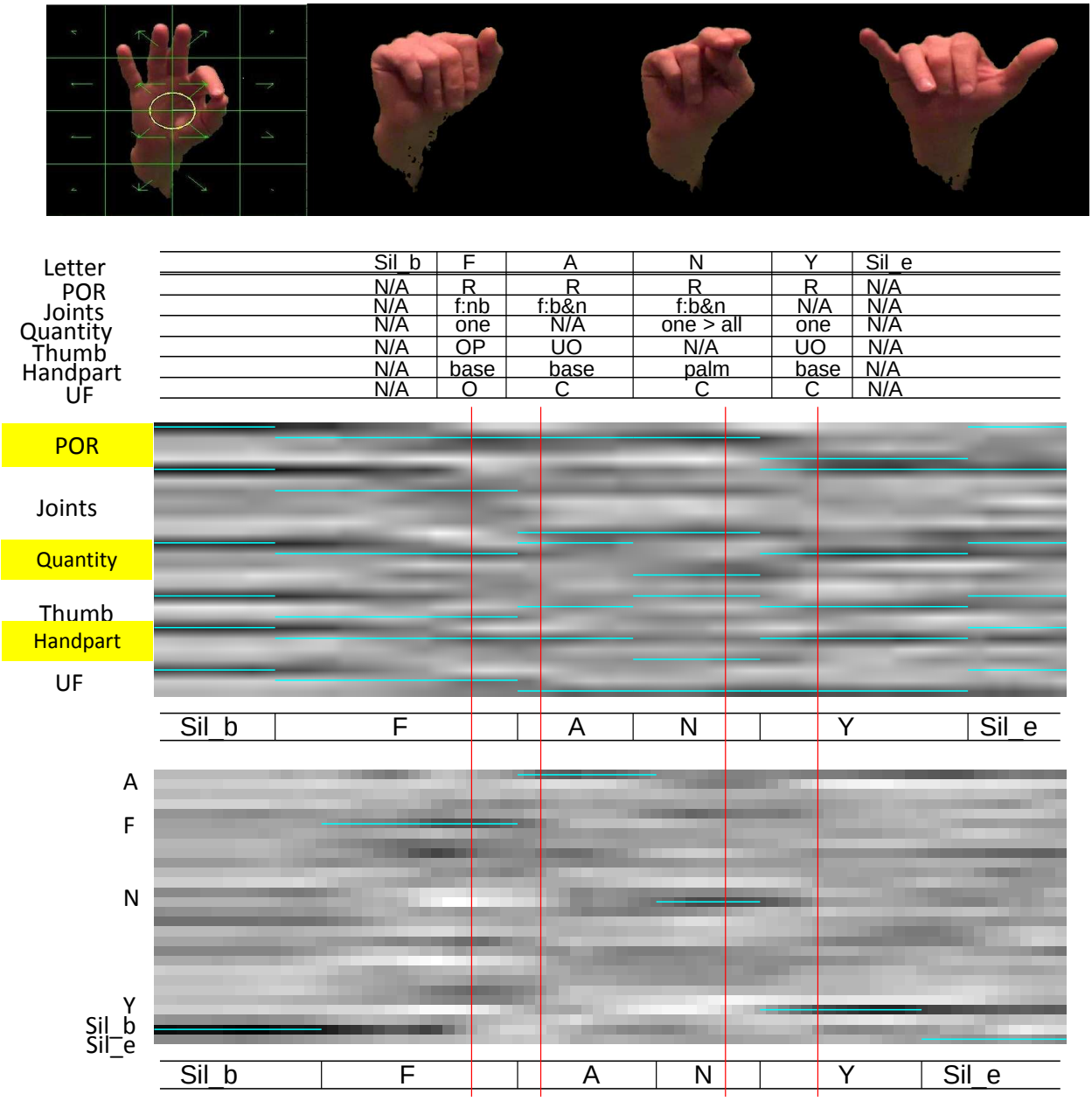

Fig. 3. Several components of fingerspelling recognition with feature- and letter-based tandem models, for an example of the sequence 'F-A-N-Y'. Top: apogee image frames of the four letters (we use all frames, but only the apogees are shown). The first image shows single-depth SIFT features (lengths of arrows show strengths of image gradients in the corresponding directions). Middle: "ground-truth" letter and feature alignments derived from the manually labeled apogees. POR = point of reference. UF = unselected fingers. Red vertical lines: manually labeled apogee times. Bottom: MLP posteriors (darker = higher) for features (upper matrix) and letters (lower matrix), and output hypotheses from feature-based and letter-based tandem models, respectively. Horizontal cyan bars: decoded hypothesis (in the feature case, the letter hypothesis is mapped to features).