Automatic Recognition of Fingerspelled Words in British Sign Language

Stephan Liwicki and Mark Everingham
School of Computing
University of Leeds
{sc06sl|m.everingham}@leeds.ac.uk

Abstract

We investigate the problem of recognizing words from video, fingerspelled using the British Sign Language (BSL) fingerspelling alphabet. This is a challenging task since the BSL alphabet involves both hands occluding each other, and contains signs which are ambiguous from the observer's viewpoint. The main contributions of our work include: (i) recognition based on hand shape alone, not requiring motion cues; (ii) robust visual features for hand shape recognition; (iii) scalability to large lexicon recognition with no re-training.

We report results on a dataset of 1,000 low quality webcam videos of 100 words. The proposed method achieves a word recognition accuracy of 98.9%.

1. Introduction

British Sign Language (BSL) is a visual language used by deaf people, which uses word level signs (gestures), non-manual features *e.g.* facial expression and body posture, and fingerspelling (letter-by-letter signing) to convey meaning. In this work we investigate the task of recognizing BSL fingerspelling from video. Fingerspelling is used to spell words for which no sign exists *e.g.* proper names or technical terms, to spell words for signs that the signer does not know, or to clarify a sign unfamiliar to the 'observer' reading the signer.

Figure 1 shows a representation of the BSL fingerspelling alphabet for a right-handed signer. BSL fingerspelling has several properties which make recognition challenging: (i) For all but one letter ("c") the letter signs involve both hands. This is in contrast to the American Sign Language (ASL) alphabet, where letters are represented by single hand shapes, and poses challenges for hand tracking and segmentation since the hands explicitly interact and occlude each other. (ii) For all but two of the letters ("h" and "j") the signs are "static" *i.e.* the meaning is conveyed by hand shape and contact points between the two hands, rather than a specific motion. In this sense the signs are not "gestures" as in word-level signs – while motion in signing a letter may give cues for recognition *e.g.* the hands must move to achieve the relevant configuration, it is a result of articulation or co-articulation (movement between signs), rather than an explicit property of the sign itself. (iii) For many signs the *pose* of the hands is also not prescribed and allowable variation makes recognition difficult. Examples are the vowels (a,e,i,o,u) which are signed by touching one of the fingertips of the left hand with the index finger of the right hand – while Figure 1 shows "caricatured" poses for these signs for the sake of clarity, any pair of hand poses which results in the defined point of contact between the fingers is allowable and encountered in practice. (iv) Finally, a number of signs – l,m,n,r,v – are difficult to recognize visually because they are distinguished only by the number and shape of the fingers laid against the palm of the hand, making segmentation challenging.

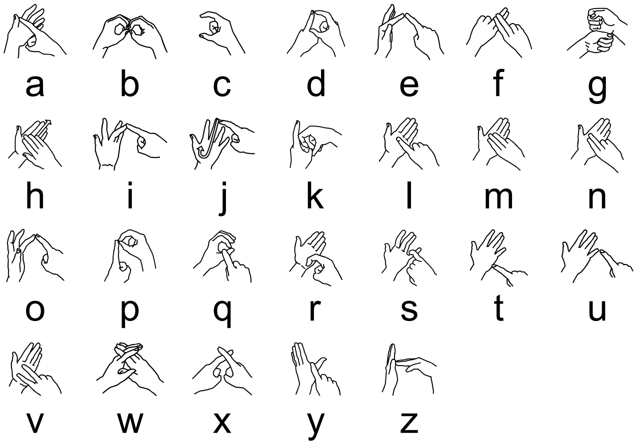


Figure 1. The BSL fingerspelling alphabet (right-handed signer). Note that the signs shown are 'caricatures' and represent poorly the imaged appearance of the sign, for example "a" is represented by the contact between right hand index finger and left hand thumb – the particular position and pose of the hands conveys no meaning in itself.

Our approach tackles these challenges in the following way: (i) we avoid attempts to explicitly track the individual hands, extracting a single appearance descriptor for the pair of hands; (ii) the method bases recognition on single image features alone. This prevents the classifier exploiting co-articulation features which vary across letter pairs, and means that we require only a small training set; (iii) variation in hand pose is overcome by the use of robust descriptors invariant to local deformation, and by training on short continuously signed sequences; (iv) implicitly ambiguous signs are disambiguated by using a lexicon of words, while not requiring re-training to expand the lexicon.

The closest related work to ours is that of Goh & Holden [9, 10] which tackles fingerspelling in Australian Sign Language (Auslan), which shares the BSL fingerspelling alphabet. A key aspect of their approach is that they focus on *dynamic* properties of the signs, using only minimal hand shape descriptors (shape moments). Signs are modelled generatively in a Hidden Markov Model (HMM) framework. The approach recognizes signs by motion between signs rather than the hand shapes. As a result, while recognition trained and tested on a small lexicon is accurate – 89% word accuracy on a lexicon of 20 words, letter-level accuracy on isolated signs is poor (57.9%) since the method learns co-articulation specific to the lexicon. By contrast, we extract detailed and robust descriptors of static hand shape and obtain high accuracy without requiring words to be represented in the training set.

Most other work has investigated the single-handed alphabets, including ASL [13, 14, 6], Korean [16] and Japanese [13]. Work in the vision domain has concentrated on hand shape descriptors and matching [13, 16] and choice of classifier [14]. Most previous methods [13, 14, 10] use skin color to segment the hands in the image, and we also adopt this approach, proposing an extension to cope with skin-colored backgrounds. To describe the shape of the hands, features of the silhouette alone are typical [13, 14, 10]. While this is reasonable for *e.g.* the ASL alphabet, the BSL fingerspelling alphabet contains signs which are ambiguous given the silhouette (l,m,n,v,r). We therefore employ a descriptor which captures internal appearance. An exception is the work of Feris *et al.* [6] which represents edges corresponding to depth discontinuities. However, their approach requires a multiple-flash camera setup [6] making it unsuitable for real-world continuous signing applications.

Overview. In Section 2 we describe the proposed method for recognition of individual signs (letters). Section 3 describes our framework for lexicon-based word recognition. Experimental results are reported in Section 4, and Section 5 present conclusions and suggests directions for future work.

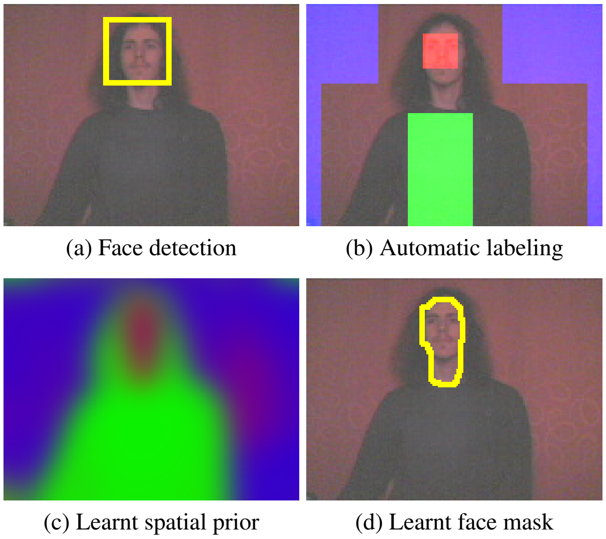


Figure 2. Bootstrapping the hand segmentation model. (a) the signer's face is detected; (b) labels are predicted for image pixels: red=face, green=clothing, blue=background; (c) a spatial prior over color models is learnt: colors indicate pixel-wise prior probability of background/clothing/face colors as RGB components; (d) a mask is learnt to exclude the face region.

2. Letter recognition

This section describes our method for individual letter recognition, consisting of hand segmentation (Section 2.1), appearance descriptor (Section 2.2) and classification (Section 2.3).

2.1. Hand segmentation

We follow much previous work in exploiting color to localize and segment the signer's hands. Other approaches to hand localization include training sliding-window detectors [15] and articulated upper-body models [4]. The latter has shown particularly impressive results for signer tracking in unconstrained TV footage [4] but is computationally expensive and requires a number of hand-segmented training frames. In contrast to some previous work which requires the signer to wear gloves or a wristband [14] to simplify localization of the hands [14, 2] or individual fingers [13], we do not place any such requirements on the signer.

Our model classifies pixels as hand or non-hand by a combination of three components: a signer-specific skin color model, a spatially-varying 'non-skin' color model, and a spatial coherence prior. Although much previous work has used generic pixel-wise skin color classification alone, this is only effective with constrained lighting [11]. The segmentation task is cast as an energy minimization problem over a Markov Random Field (MRF) [3]. Each pixel is to be assigned a label $l_i \in \{0, 1\}$ (indicating non-hand or hand) to maximize the joint probability of all labels

L conditioned on the input image I:

$$\log P(L|I) \propto \sum_i \log p(\mathbf{c}_i|l_i, \mathbf{x}_i) + \alpha \sum_{\{i,j\} \in \mathcal{N}} \psi(l_i, l_j) \quad (1)$$

The first term models the probability of an observed RGB pixel value $\mathbf{c}_i$ conditioned on the corresponding pixel label l_i and position of the pixel $\mathbf{x_i}$. The second term encourages spatial coherence in the segmentation (neighboring pixels should have the same label); a standard Ising model is used *i.e.* $\psi(l_i, l_j) = \delta_{l_i l_j}$ where δ_{ij} is the Kronecker delta function, and $\mathcal{N}$ is an 8-neighbourhood. The joint probability $P(L|I)$ is efficiently maximized using a graph-cut method [3].

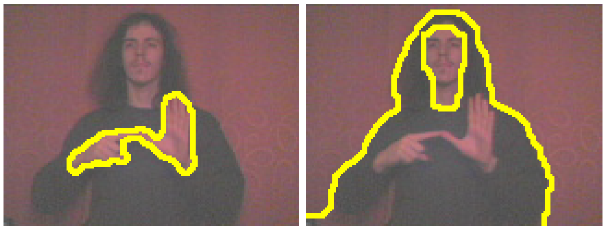
For the 'hand' label the probability of the observed pixel color is assumed independent of its position in the image *i.e.* $p(\mathbf{c}_i|l_i = 1, \mathbf{x}_i) = p(\mathbf{c}_i|l_i = 1)$. This is a pragmatic choice since it is difficult to model the variation in hand appearance due to varying pose and articulation.

For the 'non-hand' label, in contrast to most previous work [3] a *spatially-varying* color model is used. This enables accurate segmentation when the color distributions of hand and non-hand overlap considerably. Each background pixel is modeled as a mixture over three color models indexed by $j \in \{background, clothing, skin\}$:

$$p(\mathbf{c}_i|l_i = 0) = \sum_j \pi_j(\mathbf{x}_i) p(\mathbf{c}_i|j) \quad (2)$$

The mixing coefficients $\pi_j(\mathbf{x}_i)$ vary as a function of the pixel position $\mathbf{x}_i$. This captures *e.g.* that pixels in the border of the image are likely to be 'true' background, whereas pixels in the center are likely to be body-colored.

Bootstrapping the segmentation model. The proposed segmentation approach requires color models of skin, background and clothing, and the spatial background prior (2). Our method builds these *automatically* from the input video. Figure 2 illustrates the 'bootstrapping' process. We assume that for the first five frames (0.2 seconds) of video the signer remains approximately still, with their hands outside the frame. A face detector [20] is applied (Figure 2a) and, based on the face position, regions of the image likely to contain skin (red), clothing (green) and background (blue) are predicted [7] (Figure 2b). From each region a corresponding color model $p(\mathbf{c}|j)$ is estimated; we use a 3D RGB histogram with 64 bins per channel, and smooth with a Gaussian ($\sigma = 2$) to account for noise. To obtain the spatially-varying mixture coefficients each pixel is assigned its MAP label (background/clothing/skin) to give three binary masks. The masks are smoothed with a Gaussian ($\sigma = 8$) and re-normalized to have a pixel-wise sum of one. Figure 2c shows the resulting mixture coefficients with RGB components corresponding to π_j (2). Smoothing the masks is essential to model uncertainty in the position of the body as the signer changes pose – effectively the model assumes that pixels near the borders between *e.g.* background and clothing may be explained by either color model. Finally a face mask is learnt (Figure 2d) by applying the segmentation model (1) and keeping all pixels classified as skin in the bootstrap frames; this mask is used to prevent face pixels being classified as hand by constraining the corresponding pixel labels l_i.


(a) Spatial prior

(b) No prior

Figure 3. Segmentation using a spatial prior. (a) Mixing color models according to the spatial prior gives reliable segmentation; (b) A single background color model fails since clothing and background color distributions overlap with the skin color.

Example results. Figure 3a shows example results of the segmentation algorithm; further examples can be seen in Figure 10. The images are quite challenging because the skin and background color overlap considerably – this can be observed in Figure 2c, where the spatial model to the right of the signer includes a significant skin-like component. Figure 3b shows results if the spatial part of the model is removed *i.e.* $p(\mathbf{c}|l = 0)$ is modeled as a single histogram [3]; in this case segmentation fails completely since both background and clothing closely resemble skin and shadow pixels in the skin color model.

2.2. Hand shape descriptor

Given an assignment of image pixels to hand or non-hand, a descriptor is extracted to describe the hand shape. To obtain translation invariance the coordinate frame is normalized to place the centroid of the hand pixels at the origin. Note that no special steps are taken to attempt to distinguish the two hands – a single descriptor is extracted for the joint hand configuration.

The hand shape is represented by a modification of the Histogram of Oriented Gradients (HOG) descriptor [5], as a joint histogram over quantized gradient orientation and position (see Figure 4). This descriptor has recently been applied to fore-arm localization in BSL [4] and hand "grasp" recognition [12]. The intuition of the descriptor scheme is to capture local appearance of the hand configuration, while incorporating a controlled level of invariance to local deformation *e.g.* varying hand orientation or articulation.

The image gradient $\langle \frac{d}{dx}I, \frac{d}{dy}I \rangle$ is estimated by convolution (we use simple $[-1, 0, 1]$ filters [5]) and the magnitude and orientation computed. The orientation is quantized into

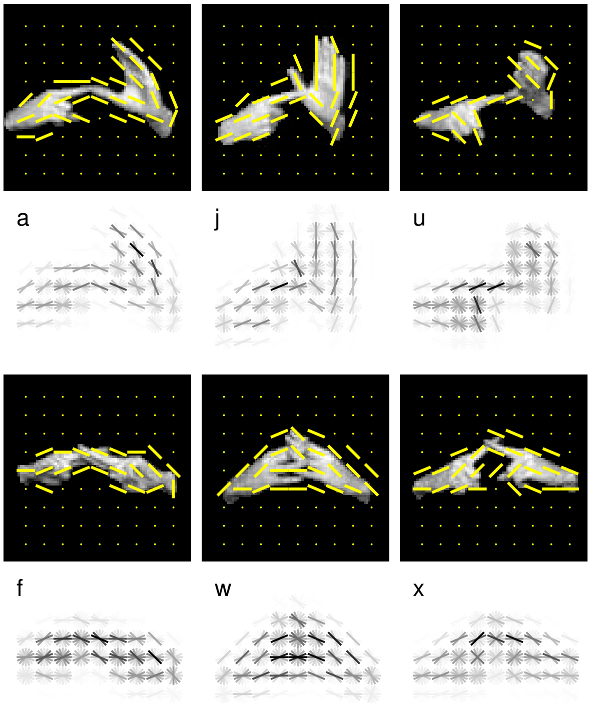


Figure 4. Hand shape descriptors. For each letter the segmented hands are shown with the dominant orientations overlaid (above) and the full orientation distribution (below) where dark bars indicate strong gradients.

a fixed number of discrete 'bins' and the gradients are then pooled into a set of spatial 'cells' by recording a histogram of quantized orientation over the image region of the spatial bin, with each orientation weighted by the corresponding gradient magnitude. In applying the descriptor to the segmented hands, only gradients within the computed hand mask are accumulated.

We apply linear interpolation between neighboring orientation bins [5] to avoid quantization artifacts, so each gradient vector contributes to two orientation bins. Spatial cells are arranged on a square grid and defined by a central point relative to the hand centroid and a fixed radius. Gaussian weighting is used when pooling orientations into each cell [5] to avoid quantization artifacts when the hand centroid is not perfectly stable. We do not apply the 'block' scheme proposed by Dalal & Triggs [5] (which renormalizes cells over larger spatial neighborhoods), as this is relevant only where there is significant local variation in contrast.

Descriptor parameters. The main parameters to be specified in the HOG descriptor are the number and spacing of cells, width of the spatial Gaussian kernel, and number of orientation bins. It is also possible to consider signed vs. unsigned gradient orientation (unsigned orientation considers light/dark vs. dark/light transitions equivalent). All parameters were set by cross-validation on the training data (see Section 4); the best results were obtained with a 9×9 grid of cells spaced at 8 pixels, a spatial Gaussian with $\sigma = 4$, and 8 unsigned orientation bins.

Example descriptors. Figure 4 shows visualizations of the hand shape descriptor for two sets of letters whose signs have similar silhouettes. For each sign the top image shows the segmented input with the dominant gradient orientation in each spatial cell overlaid. The lower image shows the complete descriptor, with the magnitude in each spatial/orientation bin encoded by gray-level (darker is stronger).

The fairly coarse spatial cells and orientation bins give some invariance to varying global and relative orientation of the hands, while still retaining discriminative features. It can be seen that the descriptor also captures internal features of the hand shape, not merely the silhouette shape. This is particularly visible for the example of "w" (Figure 4, last row) where the 'interlacing' of the fingers is clearly visible in the descriptor, and can be distinguished from the crossed index fingers for "x" despite the overall similarity in the silhouette. We evaluate the importance of capturing such internal features in Section 4.

2.3. Classification

Given a hand shape descriptor $\mathbf{z}$, we wish to assign it a class $C \in \{a, \ldots, z, \omega\}$ where ω denotes 'non-letter' (see Section 3). Since the descriptor is a fixed-length vector, a variety of standard multi-class classification schemes are applicable. We investigated both linear and kernelised classifiers. A good compromise between accuracy and computational efficiency was found to be multi-class logistic regression (see [1]). Logistic regression defines the posterior probability of a class in terms of *linear* functions of the descriptor vector:

$$P(C_k|\mathbf{z}) = \frac{\exp \mathbf{w}_k^T \mathbf{z}}{\sum_j \exp \mathbf{w}_j^T \mathbf{z}} \tag{3}$$

While the discriminative power of linear discriminants might be expected to be limited, the dimensionality of our descriptor is high (648-D) which empirically reduces the need for more complex discriminants [5].

Probability estimates. In order to combine the single-letter classification with contextual information *e.g.* a lexicon model (Section 3) it is important to have good posterior probability estimates, rather than a single class prediction. The multi-class logistic regression approach is particularly applicable here as it directly estimates class posteriors (3); this is in contrast to *e.g.* typical multi-class support vector machine (SVM) schemes. We found that introducing a

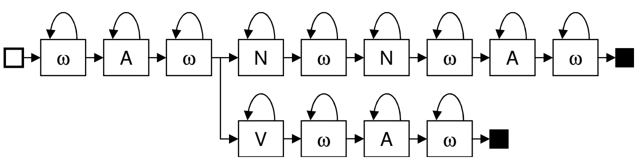


Figure 5. Hidden Markov model representation of the lexicon. Boxes denote states tagged with corresponding letters ("ω"=non-letter). Hollow and filled squares indicate start/end states. Paths from start to end states correspond to words (Anna/Ava). Probabilities are assigned to arcs for self-transitions and corresponding letter/non-letter transitions.

kernel gave marginally better accuracy for individual letter recognition but did not improve word recognition, and using a support vector machine calibrated for probability outputs gave poor results – this is no doubt significantly due to the limited training data available to calibrate the classifier's probability estimates.

Classifier parameters. The parameters $\mathbf{w}$ are set by maximum likelihood on the training data (see [1]). We add a regularization term which penalizes large weights $\mathbf{w}$ in terms of their L2 norm, and set the strength of this term by cross-validation on the training data.

3. Word recognition

The single letter classifier outputs a posterior distribution over letters and non-letter for a single image. To recognize continuously signed words the classifier is combined with a lexicon of known words, modeled as a Hidden Markov Model (HMM) [17]. The aims are twofold: (i) to suppress the classifier output over multiple frames *e.g.* repeated output for a single sign; (ii) to disambiguate signs which are difficult to distinguish visually.

A HMM is built for each word in the lexicon as a chain of alternating letter and non-letter states. For computational efficiency the HMMs for all words are combined in a tree structure – the model for a particular word forms a single path through the tree. Figure 5 shows a small part of the tree for two words – "Anna" and "Ava".

The probability of a word w given the sequence of observed descriptors $\mathbf{z}_{1\ldots k}$ is defined as

$$P(w|\mathbf{z}_{1\ldots k}) \propto \max_{s_1,\ldots,s_k} P(s_1) \prod_{t=1}^{k} p(\mathbf{z}_t|s_t) \prod_{t=2}^{k} P(s_t|s_{t-1}) \tag{4}$$

where s_t denotes the (unobserved) state at time t (one of the nodes in Figure 5). The maximization over the state sequence is efficiently performed using the Viterbi algorithm [17] in time $O(nk)$ where n is the number of nodes in the lexicon tree. A given sequence is then assigned a predicted word by finding which path from start to end states maximizes the posterior $P(w|\mathbf{z}_{1\ldots k})$.

Use of posterior probabilities. Since there is a one-to-one correspondence between unobserved states of the model and letters, the 'emission' probabilities $p(\mathbf{z}_t|s_t)$ are easily defined as $p(\mathbf{z}_t|\lambda_t)$ where λ_t is the letter (or non-letter) corresponding to state s_t (labels in boxes of Figure 5). Note however, our classifier instead outputs posterior probabilities $P(\lambda_t|\mathbf{z}_t)$. It is straightforward to show that these differ only by an unknown time-dependent normalization factor *i.e.* $P(\lambda_t|\mathbf{z}_t) = \frac{1}{Z_t} p(\mathbf{z}_t|\lambda_t)$. Since the normalization factor is not dependent on state, it is permissable to substitute our (probabilistic) classifier outputs.

HMM parameters. The proposed HMM requires just two parameters to be specified which define the transition probabilities $p(s_t|s_{t-1})$ (4): the self-transition probabilities for letter and non-letter states. Lacking further information we assume the probabilities to be equal for all letters in the alphabet *i.e.* that each sign is of approximately equal duration. The parameters were set on a hold-out set and results are not critical on their setting; we use $P_{self}(letter) = P_{self}(nonletter) = 0.9$.

Relation to HMMs for gesture recognition. It is important to note that the proposed HMM differs from most work on *gesture* recognition [21, 19] in that the HMM for a word can be built *without* examples of that word, making the approach scalable to large lexicons. This is because of the one-to-one correspondence between (unobserved) states and desired output (letters). This is in contrast to *e.g.* previous work on BSL fingerspelling [10] where examples of whole words to be recognized are required to learn both letter and word recognition models.

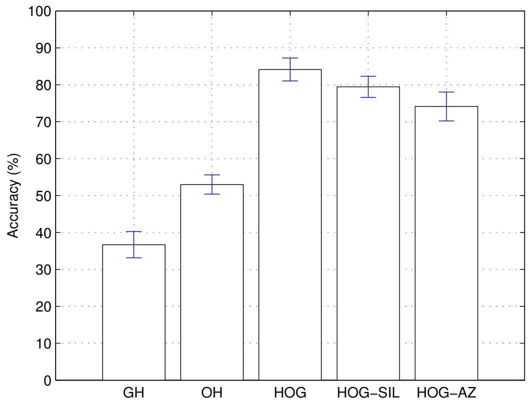
4. Experimental results

This section reports results of our proposed method in terms of single letter and continuous word recognition.

Dataset. All experiments reported here use a newly-collected dataset with test data consisting of 1,000 videos of 100 words (10 repetitions) signed by a single novice signer. The 100 words chosen are the most common male and female forenames in the USA, 2000–2007 [18][1]. Forenames are particularly relevant for this task as they are commonly fingerspelled. They also present challenges because of very similar words *e.g.* alexander/alexandra both appear in the dataset. The video was captured in a home environment without any special lighting, using a consumer-quality USB webcam. The videos are low-resolution (160 × 120 pixels) and MJPEG compressed.

Two small sets of training data were used, disjoint from the test set: (i) three repetitions of two (arbitrarily chosen)

[1] The irony of using US names with BSL does not escape us, but we are unaware of a comparable British list!



Method	Mean (%)	SD (%)
GH	36.7	3.5
OH	53.0	2.6
HOG	**84**.1	**3.1**
HOG-SIL	79.5	2.9
HOG-AZ	74.1	3.9

Figure 6. Single letter recognition accuracy for baseline and proposed descriptors. Error-bars indicate ± 1 standard deviation over five samples of 200 signed letters.

pangrams (sentences containing every letter in the alphabet): "The quick brown fox jumps over the lazy dog" and "Six big devils from Japan quickly forgot how to waltz"; (ii) three repetitions of the letters of the alphabet signed individually. The idea of using pangrams is to capture 'natural' hand shapes since signs are performed in the context of words – Section 4.1 reports a comparison with alphabet data. For the 'dynamic' signs ''j" and "h" training frames are taken from the middle and end of the sign respectively.

4.1. Single letter recognition

We first report results on recognition of individual letters. The test data contains 5 independent sets of 200 letter signs, sampled randomly from the test set according to the overall distribution of letters in the 100 words.

Baseline descriptors. We implemented two baseline descriptors to compare against the proposed HOG descriptor. (i) "GH" is the shape descriptor used in the method of Goh & Holden [10]: four features of the binary hand mask are computed from the covariance matrix of pixel coordinates within the mask: area, length of major and minor axes, and orientation; (ii) "OH" implements the orientation histogram proposed for hand gesture recognition by Freeman & Roth [8]. This descriptor represents hand shape by a global orientation histogram, and is approximately equivalent to a HOG descriptor with a single spatial cell.

Descriptor comparison. Figure 6 reports letter recognition accuracy for the proposed (HOG) and baseline descriptors. The accuracy is class-balanced so chance performance (random guess) is $1/26 \approx 3.8\%$. In all cases the classifier is multi-class logistic regression (Section 2.3) and parameters were chosen by cross-validation on the training set.

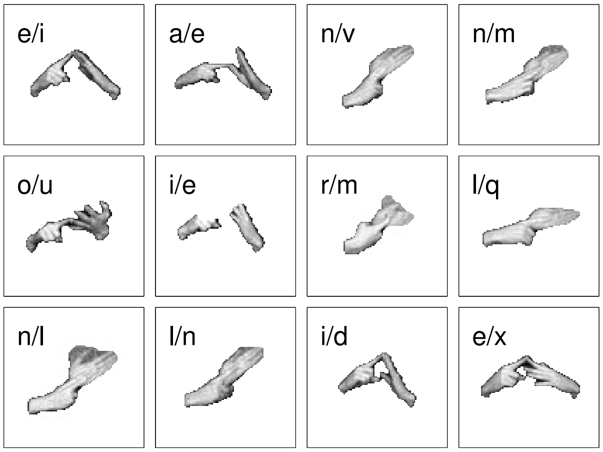


Figure 7. Letter recognition errors. Examples are shown of the most frequently confused letters. Labels x/y indicate ground truth letter x and (erroneous) prediction y.

The GH descriptor [10] performs poorly, giving accuracy of 36.7%; this is unsurprising given the very simple representation of silhouette alone. The OH descriptor [8] gives accuracy of 53.0%; for this descriptor 16 *signed* orientation bins performed best – using signed orientation gives the descriptor some limited ability to represent spatial information as the background is reliably darker than the hand. The proposed HOG descriptor gives the highest accuracy by a significant margin: 84.1%.

Use of internal features. To establish the influence of the proposed descriptor's ability to make use of internal features rather than the hand silhouette alone, we computed HOG over a silhouette image. We also investigated smoothing the silhouette before descriptor computation to reduce gradient artifacts caused by use of a binary image. When using the silhouette alone (HOG-SIL in Figure 6) accuracy reduces from 84.1% to 79.5%, showing that the proposed descriptor benefits from the representation of internal features.

Pangram training data. We argue that using pangrams as training data is an effective way to capture natural variation in signs without onerous training requirements (*e.g.* word-level [10]). We validated this by comparing accuracy trained on individually signed letters of the alphabet (HOG-AZ). The accuracy reduced considerably from 84.1% (pangrams) to 74.1% (alphabet) showing the advantage of the pangram training data.

Confused letters. Figure 7 shows examples of the most frequently confused letters (Figure 1 is a useful reference

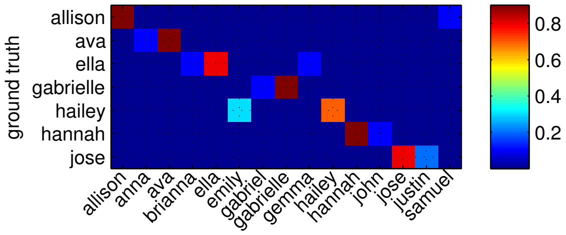


Figure 8. Confusion matrix for word recognition on the 100-name dataset. Entries are shown only for the 7 words with classification accuracy less than 100% (rows). Columns show the correct/incorrect predicted word.

here). Most errors can be assigned to two sets: (i) nearby vowels (e/i) – these are difficult to distinguish if the signer adopts an 'unhelpful' hand pose; (ii) 'palm-based' signs (l/m/n/r/v) – although many examples are correctly classified the combination of low resolution and varying hand pose makes it very difficult to distinguish the number and orientation of fingers laid against the palm. Other miscellaneous errors arise from segmentation errors (i/e), or coincidental hand poses which closely resemble other signs (i/d and e/x).

4.2. Word recognition

To test the proposed model for recognition of continuously-signed words (Section 3) the full test set of 1,000 instances of 100 words was used. Note that the training data consists only of single letter/non-letter examples (not words). The non-letter class is treated identically to the letter classes, with non-letter examples sampled randomly from the training videos. Word recognition accuracy is 98.9% (a total of 11 out of 1,000 sequences are misclassified). We consider this very promising given the large lexicon, and that we use no examples of words in the lexicon as training data. For comparison, Goh & Holden [10] report word accuracy of 88.6% on a lexicon of 20 words, and require 4 instances of each word for training. Figure 10 shows some examples of correctly recognized words and the frames identified as the corresponding letters.

Confused words. Figure 8 shows the confusion matrix for the seven words (11 sequences) which were sometimes misclassified. Some errors can be anticipated and are caused by ambiguity in vowels and 'palm-based' signs *e.g.* ava/anna and ella/gemma. Visual inspection of the other errors reveals errors in the segmentation to be the main cause.

Scalability to large lexicons. While we consider our results on a lexicon of 100 words impressive, it is salient to consider what accuracy is likely to be achieved with larger lexicons. Lacking test data of more than 100 words, we present an approximation here. We increased the lexicon size from 100 up to 2,000 words by sampling additional words from the same list of popular forenames from which the original lexicon was taken. We then run word recognition experiments on the original 100 words – the additional lexicon entries represent 'distracter' possibilities which the word recognizer should not predict.

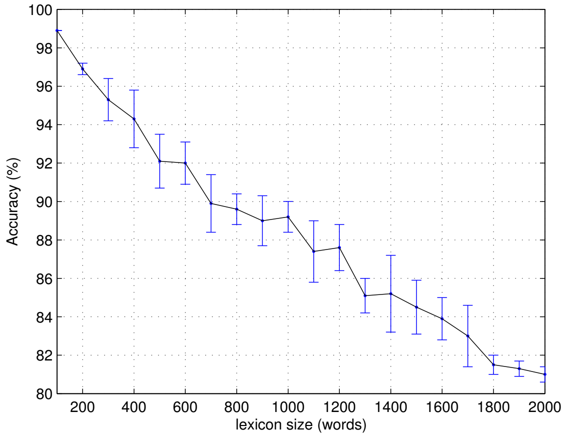


Figure 9. Word recognition accuracy as a function of lexicon size. Error-bars indicate ±1 standard deviation. See text for details.

Figure 9 shows the results. Accuracy can be seen to steadily decline with lexicon size: increasing from 100 words to 200 reduces the accuracy from 98.9% to 95.3%; for 1,000 words the accuracy is 87.4%, and is 81.0% for 2,000 words. While perhaps disappointing, these results predict our method would still out-perform previous work [10] for a lexicon of 900 words compared to just 20. These results might also be considered 'pessimistic' in that forenames are often very similar due to *e.g.* masculine/feminine forms *e.g.* alex/alexa, however as noted this domain is highly relevant to sign language. The conclusion we take from this experiment is that future work should concentrate on sign recognition rather than more complex language models – the latter are unlikely to boost real-world performance. Note that models of word sequences are not appropriate since fingerspelling is mostly used to sign single words.

5. Conclusions and future work

We have demonstrated that combining a state-of-the-art appearance descriptor with a simple HMM-based lexicon model can give highly accurate fingerspelling recognition on a large lexicon. The proposed method has an advantage over previous work in not requiring word-level training, making it scalable, and we showed that pangrams are a useful source of natural training signs. In improving the method, our results suggest that work should focus on letter-level recognition rather than prior language models. It seems promising to investigate extracting cues from multiple frames *e.g.* 'early' formation of the hand shape, without

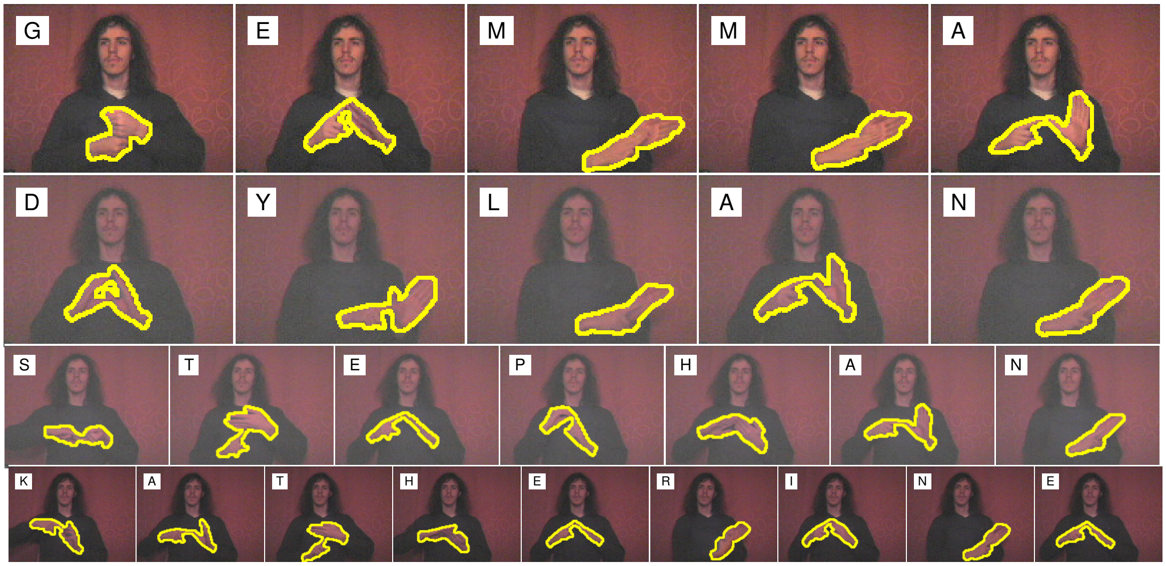


Figure 10. Example word recognition results. For each word (Gemma, Dylan, Stephan, Katherine) one frame for each recognized letter is shown, with the segmentation overlaid. The full videos are 60–120 frames (2–4 seconds) in length.

full gesture modeling which compromises scalability.

Limitations of our work are that our dataset currently contains only a single inexperienced signer, and that the imaging conditions are only moderately challenging, compared to *e.g.* signing of TV footage [4]. We hope that expanding the training data with other signers will remove the need for signer-specific training, and aim to investigate "front end" methods robust enough to deal with arbitrary and dynamic imaging conditions.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

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

[3] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*, pages 105–112, 2001.

[4] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In *Proc. BMVC.*, 2008.

[5] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. CVPR*, pages 886–893, 2005.

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

[7] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR*, 2008.

[8] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *In Proc. Int. Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1995.

[9] P. Goh. Automatic recognition of Auslan finger-spelling using hidden Markov models. Undergraduate thesis, University of Western Australia, 2005.

[10] P. Goh and E.-J. Holden. Dynamic fingerspelling recognition using geometric and motion features. In *Proc. ICIP*, pages 2741–2744, 2006.

[11] M. Jones and J. Rehg. Statistical color models with application to skin detection. *IJCV*, 46(1):81–96, 2002.

[12] H. Kjellstrom, J. Romero, D. Martinez, and D. Kragic. Simultaneous visual recognition of manipulation actions and manipulated objects. In *ECCV*, pages 336–349, 2008.

[13] M. Lamari, M. Bhuiyan, and A. Iwata. Hand alphabet recognition using morphological pca and neural networks. In *Proc. IJCNN*, pages 2839–2844, 1999.

[14] R. Lockton and A. W. Fitzgibbon. Real-time gesture recognition using deterministic boosting. In *Proc. BMVC.*, pages 817–826, 2002.

[15] E. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Proc. Int. Conf. FG*, 2004.

[16] A. Park, S. Yun, J. Kim, S. Min, and K. Jung. Real-time vision-based Korean finger spelling recognition system. *Proc. World Academy of SET*, 34:293–298, 2008.

[17] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989.

[18] Social Security Online. Popular baby names of the 2000s. `www.ssa.gov/OACT/babynames`.

[19] T. Starner, J. Weaver, and A. Pentland. Real-time American sign language recognition using desk- and wearable computer-based video. *PAMI*, 20(12):1371–1375, 1998.

[20] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 57(2):137–154, 2004.

[21] C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *Proc. ICCV*, pages 363–369, 1998.