Minimal Training, Large Lexicon, Unconstrained Sign Language Recognition

Timor Kadir[2], Richard Bowden[1,2], Eng Jon Ong[1] and Andrew Zisserman[2]
[1]CVSSP, University of Surrey, Guildford, UK
{r.bowden,e.ong}@surrey.ac.uk
[2]Dept Engineering Science, University of Oxford, Oxford, UK
{timork,az}@robots.ox.ac.uk

Abstract

This paper presents a flexible monocular system capable of recognising sign lexicons far greater in number than previous approaches. The power of the system is due to four key elements: (i) Head and hand detection based upon boosting which removes the need for temperamental colour segmentation; (ii) A body centred description of activity which overcomes issues with camera placement, calibration and user; (iii) A two stage classification in which stage I generates a high level linguistic description of activity which naturally generalises and hence reduces training; (iv) A stage II classifier bank which does not require HMMs, further reducing training requirements.

The outcome of which is a system capable of running in real-time, and generating extremely high recognition rates for large lexicons with as little as a single training instance per sign. We demonstrate classification rates as high as 92% for a lexicon of 164 words with extremely low training requirements outperforming previous approaches where thousands of training examples are required.

1 Introduction

The objective of this paper is to efficiently and accurately recognise signed words, from British Sign Language, using a minimal number of training examples. Furthermore, our aim is to use natural image sequences, without the signer having to wear data gloves or coloured gloves to aid the visual processing, and to be able to recognise hundreds of signs. The motivation for this work is to provide a real time interface so that signers can easily and quickly communicate with non-signers.

Drawing on the analogy with speech recognition, previous approaches to sign language recognition have extracted feature vectors from the image and then fitted a Hidden Markov Model (HMM) [5, 9], and carried out extensive training to extract the important temporal transitions for classification. This requires large amounts of data to represent the variation in events, feature description and any co-articulation that occurs. We, on the other hand, extract a higher level description which performs this generalisation. We then represent the temporal transitions using a Markov chain, that is, there is no need for an enforced hidden layer. This avoids the high training requirements (of HMMs) and provides a scalable solution.

BMVC 2004 doi:10.5244/C.18.96

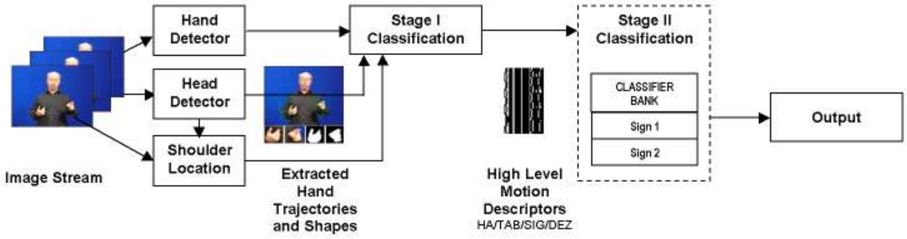


Figure 1: Block diagram showing a high level overview of the stages of classification.

In previous work, Volgar used 1292 training examples for a lexicon of 22 signs or 2345 for 53 signs [9]. Starner used 1920 training examples (or 2000 depending upon the test) for a 40 sign lexicon[5]. These HMM approaches required 40-100 individual training examples of each sign. As will be seen, our approach can perform equally well on similar size lexicons with as little as a single example per sign. As the number of examples required is dependent upon the size of the lexicon (i.e. bi-sign co-articulation) it is evident that with a HMM approach it is not feasible to address large lexicon recognition.

Following our previous work [1], the strength of our approach is that we structure the classification model around a linguistic definition of signed words. This enables signs to be learnt reliably from just a single training example, and we have been able to reach 164 (so far) with only a handful of examples per sign.

The remainder of this paper details the system and its performance. Following a systems overview, Section 3 provides details of the AdaBoost classifier used to detect both head and hands. We then present details of our 2 stage classification approach in Section 4. Finally results are presented in Section 5 and conclusions drawn.

2 Overview

A graphical overview of the system is given in Figure 1. The position of the head and hands is first extracted from the video sequence using two classifier cascades trained using boosting techniques. The position of the head is then used to locate the shoulders and from this the approximate position of the torso and its constituent parts are gained. This information is then passed to the stage I classification.

Classification stage I: This initial stage of classification converts the hand motion into a phonetic (or viseme) representation taken from sign linguistics [2]:

HA	Position of the hands relative to each other
TAB	Position of hands relative to key body locations
SIG	Relative movement of the hands
DEZ	The shape of the hand(s)

This HA/TAB/SIG/DEZ notation provides a high-level feature descriptor that specifies events in broad terms such as *hands move apart, hands touch or right hand on left* *shoulder*. This broad description of scene content naturally generalises temporal events and hence reduces training requirements. This is described in more detail in Section 4.

Classification stage II: Each sign is modelled as a 1st order Markov chain where each state in the chain represents a particular set of feature vectors from the stage I classification. The Markov chain encodes temporal transitions of the signer's hands. During classification, the chain which produces the highest probability of describing the observation sequence is deemed to be the recognised word.

3 Detecting Faces and Hands using Boosted Classifiers

Boosting is a general method that can be used to improve the accuracy of a given learning algorithm [4]. More specifically, it is based on the principle that a highly accurate or 'strong' classifier can be produced through the linear combination of many inaccurate or 'weak' classifiers. Formally, given an image ($\mathbf{x}$), a strong classifier layer ($W_i(\mathbf{x})$) can be defined as a linear combination of the outputs of a number (M_i) of weak classifiers ($w_{i,m}$):

$$W_i(\mathbf{x}) = \sum_{m=1}^{M_i} w_{i,m}(\mathbf{x}) \tag{1}$$

Here, we require the output of W_M to be positive for detection and zero or negative for no detection.

Classifier efficiency is increased by organising the weak classifiers into a 'cascade' [8], where the number of weak classifiers in each layer of the cascade increases with depth. The purpose of this is that initial layers (which are required to test many possible hypotheses) are simple to compute. They should reject large numbers of hypotheses such that later layers (which are more complex and therefore computationally expensive) need only be applied to a small subset of the original hypotheses. In this fashion an exhaustive search over all positions and scales is possible as in excess of 90% of possible hypotheses can be rejected at each stage of the cascade.

A weak classifier is a function which, given an image as input, returns a detection value. We have chosen to use the Harr wavelet-like features as used by Viola and Jones [8] and Li *et al.* [10]. These are disjointed block differences, and may be computed efficiently using an integral image[8].

Two probability densities of block difference values are then built to produce the weak classifier. Formally, the i^{th} weak classifier belonging to the m^{th} cascade layer can be defined as:

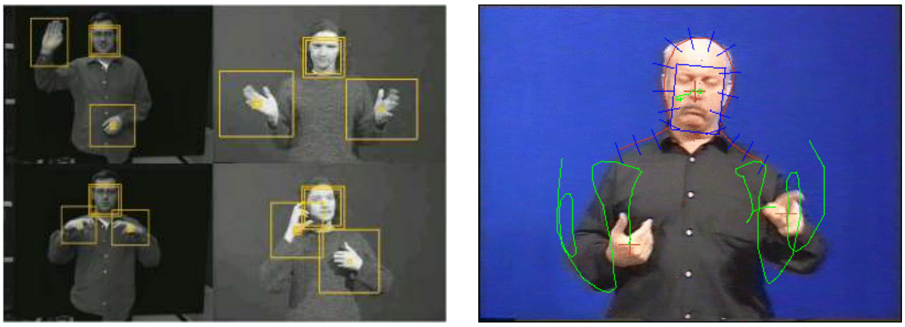
$$w_{i,m}(x) = 0.5(\frac{p_{i,m}(z|y=+1,w)}{p_{i,m}(x|y=-1,w)} - T_{i,m}) \tag{2}$$

where w are the set of weights associated with each example during classifier construction, z is the value of the block differences, and T is the threshold that determines how easy it would be for the classifier to classify positive examples. For each strong classifier (the collection of weak classifiers in any given layer of the cascade) the exact collection of weak classifiers is chosen using the AdaBoost algorithm [3], such that an exponential loss function modelling the upper bound of the training error is used:

$$J(W_i) = \sum_{j}^{N_W} e^{-y_j W_i(\mathbf{h}_j)} \tag{3}$$

Model	No. Layers	Cascade Layers	+ve Training Eg	-ve Training Eg
Head	10	2,5,5,20,50,50,150,150,150,150	2500	6000
Hands	8	2,5,5,20,50,50,150,150	2400	6000

Table 1: Cascade Classifier Specifications.



(a)

(b)

Figure 2: (a) Example detections from the cascade head and hand detectors. (b) The result of contour tracking and trajectory of the hands over time.

were a training set ($\mathbf{h}_j$) of (N_W) images consists of objects and non-objects. The objects and non-object images are associated with labels (defined as y_i) +1 and -1 respectively. Weak classifiers are added sequentially into an existing set (W_i) of layer (M_i) such that this upper bound is decreased.

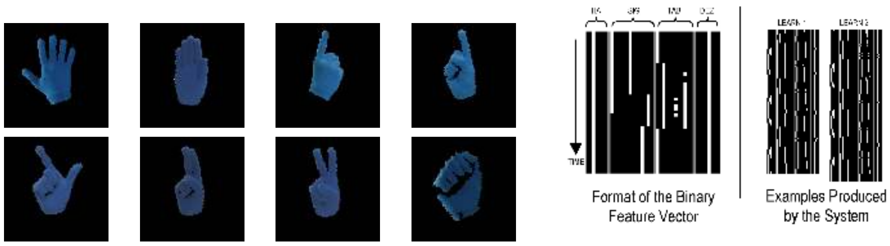
We train two different strong classifiers for the head and hands respectively, defined in details in Table 1. Detection proceeds by searching the entire image over all positions and scale. For each particular section of an image, the weak classifiers are transformed such that they apply only to that image section. If the image content does not contain the object of interest (head or hands), it will be rejected early in the cascade, otherwise it will filter down to the final layer and be accepted. The strongest detected head and hand hypotheses are then passed to stage I classification. Figure 2a shows some sample images taken from 2 sequences with the detected location of head and hands outlined.

4 Two Stage Classifier

The robust detection of the head (Section 3) provides a good indication of the position and scale of the signer in the image. From this estimate, a 2D contour model of the head/shoulders is fitted to high magnitude gradients around the signer. The gradients are determined by convolving along normals with a 1D, 2nd derivative Gaussian kernel. The contour is a coarse approximation to the shape of the shoulders and head consisting of 18 points connected together (see Figure 2b) and is a mathematical mean shape taken from a number of sample images of people. The contour is fitted to the image by estimating the local similarity transformation which minimises the contour's distance to local image features.

HA	TAB	SIG	DEZ
1. Right hand high	1. The neutral space	1. Hand makes no movement	1. 5
2. Left hand high	2. Face	2. Hand moves up	2. A
3. Hands side by side	3. Left Side of face	3. Hand moves down	3. B
4. Hands are in contact	4. Right Side of face	6. Hand moves left	4. C
5. Hands are crossed	5. Chin	7. Hand moves right	5. F
	6. R Shoulder	8. Hands moves apart	6. G
	7. L Shoulder	9. Hands move together	7. H
	8. Chest	10. Hands move in unison	8. I
	9. Stomach		9. P
	10. Right Hip		10. V
	11. Left Hip		11. W
	12. Right elbow		12. Y
	13. Left elbow		

Table 2: The high level features after stage I classification.


HA
SIG
TAB
DEZ
TIME
Format of the Binary Feature Vector
LEARN 1
LEARN 2
Examples Produced by the System

(a) (b)

Figure 3: (a) Examples of hand-shapes used in British Sign Language – from top-left clockwise ‘5’, ‘B’, ‘G’, ‘G’, ‘A’, ‘V’, ‘H’, ‘G’. (b) Graphical representation of feature vectors for different occurrences of signs demonstrating the consistency of feature description produced and two instances of the sign ‘learn’.

Estimates for key body locations (such as shoulders, chest, hips etc.) are placed relative to the location of the head contour. This provides a body centred co-ordinate system with respect to which the position and motion of the hands is described in terms of the HA,TAB and SIG feature descriptors. This means that as the contour is transformed to fit the location of the user within the video stream, so the approximate locations of the key body components are also transformed. Figure 2b shows the system tracking the upper torso and hands of an active signer, the trails from the hands show the path taken over the last 50 frames.

The 5 HA, 13 TAB, 10 SIG and 12 DEZ states we currently use are listed in Table 1 and are computed as follows:

HA: the relative position of the hands to each other is derived directly from deterministic rules on the relative x and y co-ordinates of the centroids of the hands and their approximate area in pixels.

TAB: the position of the hands is categorised in terms of their proximity to key body

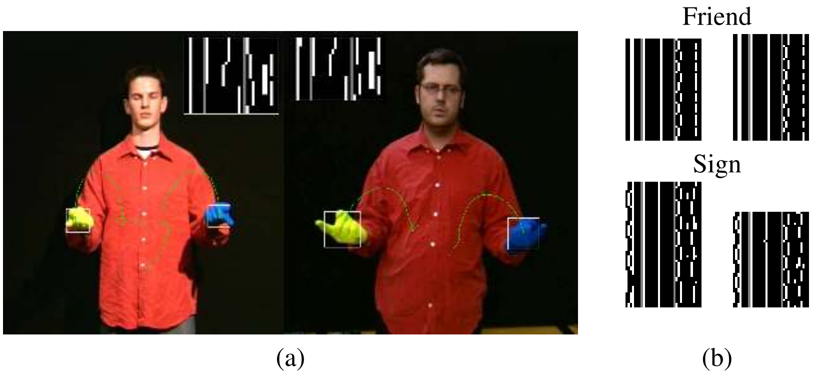


Figure 4: Generalisation of feature vectors across different individuals: (a) two people signing the sign ‘different’; (b) binary feature vectors for two people for the signs ‘friend’ and ‘sign’. The signs are correctly classified.

locations using the Mahalanobis distance computed from the approximate variance of these body parts gained from contour location.

SIG: the movement of the hands is determined using the approximate size of the hand as a threshold to discard ambient movement and noise. The motion is then coarsely quantised into the 10 categories listed in Table 1.

DEZ: British Sign Language has 57 unique hand-shapes (excluding finger-spelling) which may be further organised into 22 main groups [2]. A visual exemplar approach is used to classify the hand shape into twelve (of the 22) groups. This is described in detail below.

Figure 3b shows examples of the features generated by the system over time. The horizontal binary vector shows HA, SIG, TAB and DEZ in that order delineated by grey bands. The consistency in features produced can clearly be seen between examples of the same word. It is also possible to decode the vectors back into a textual description of the sign in the same way one would with a dictionary. The feature vector naturally generalises the motion without loss in descriptive ability. Figure 4 shows the word ‘different’ being performed by two different people along with the binary feature vector produced. The similarity is clear, and the signed words are correctly classified.

Linguistic evidence [7] points to the fact that sign recognition is primarily performed upon the dominant hand (which conveys the majority of information) we therefore currently discard the non dominant hand and concatenate HA, TAB, SIG and DEZ features together to produce a 40 dimensional binary vector which describes the shape and motion in a single frame of video.

4.1 Hand shape classification

Within the lexicon of 164 words used in this work, twelve of the main sign groups appear, denoted ‘5’, ‘A’, ‘B’, ‘C’, ‘F’, ‘G’, ‘H’, ‘I’, ‘P’, ‘V’, ‘W’ and ‘Y’ following the definition in [2]. Typical examples of these are shown in figure 3. Our objective here is to classify hand-shapes into one of these 12 groups.

The visual appearance of a hand is a function of several factors which a successful hand shape classifier must take into account. These factors include: pose, lighting, occlusions and intra/inter-signer variations. To deal with this variation we adopt an exemplar based approach, where many visual exemplars correspond to the same sign group.

Segmentations of the hands are first obtained from the tracking stage discussed in Section 3 using a colour based segmentation. Hand-shape is represented as a binary mask corresponding to the silhouette of the hand and these masks are normalised for scale and orientation using their first and second moments. Learning proceeds by generating a set of normalised masks from training data (see Section 5) and clustering these to form an exemplar set. We use a greedy clusterer with a normalised correlation distance metric. A threshold on this distance controls the degree of grouping.

Novel hand-shapes are then classified by matching their normalised binary masks to the nearest one in the exemplar set using normalised correlation as a distance metric. Similar approaches have been used widely, for example [6].

This simple exemplar approach has a number of attractive properties. Different hand-shapes in the same group, and variations in appearance of the same hand-shape due to different poses or signer variation, may be represented by separate exemplars assigned to the same group label.

While it is clear that this basic hand classifier cannot distinguish between all poses and shapes, we demonstrate that it complements the HA, TAB and SIG features and hence is *sufficient* to discriminate between many of the signs in our lexicon. Set at an operating point chosen to give 1515 exemplars, the hand-shape classifier achieves an average correct classification rate of 75%. The results presented in Section 5 use this operating point.

4.2 Stage II Training

In order to represent the temporal transitions which are indicative of a sign we make a 1st order assumption and construct a 1st order Markov chain for each word in the lexicon. However, this assumption requires that an ergodic model is constructed. With a 40 dimensional binary feature vector this would result in a chain with $2^{28} \times 12$ states (5 HA + 13 TAB + 10 SIG multiplied by the 12 mutually exclusive DEZ features) and over 10.3×10^{18} possible transitions requiring a prohibitive amount of a storage. However, as can be seen in Figure 3b, the number of transitions in each word is typically small and governed by the duration of the sign and the capture rate (in this case 25Hz).

It is also evident that out of the $2^{28} \times 12$ possible states only a small subset will ever occur with even fewer transitions, due to the physical limitations of the human body. Therefore, we build only as much of the ergodic model as is required. This is done by adding new symbols to the state transition matrix as they occur during training using a look up table (LUT). The result is a sparse state transition matrix, $P_w(s_t|s_{t-1})$, for each word w giving a classification bank of Markov chains.

4.3 Stage II Classification

During classification, the model bank is applied to incoming data in a similar fashion to HMMs. The objective is to calculate the chain which best describes the incoming data i.e. has the highest probability that it produced the observation sequence s.

First, symbols are found in the symbol LUT using an L1 distance on the binary vectors. Strictly speaking, such a distance metric is not valid in the binary feature space but

HA/TAB/SIG					
No. Training Examples	1	2	3	4	5
Mean	65.3	73.6	77.2	78.7	79.2
Minimum	55.8	71.5	75.1	75.9	76.1
Maximum	68.5	75.2	80.3	81.6	82.4
Std. Deviation	3.5	1.4	1.5	1.9	2.1
HA/TAB/SIG/DEZ					
Mean	76.2	84.9	87.0	89.5	89.1
Minimum	70.5	81.9	84.5	88.2	86.2
Maximum	79.4	87.0	88.9	90.4	92.0
Std. Deviation	2.5	1.5	1.6	0.7	1.8

Table 3: Classification performance. Note the improvement when the handshape information (DEZ) is added.

it seems to allow some correction of small errors in the vectors, which on balance results in a useful improvement in classification.

Symbols that were not present in the training sequence are added to the symbol LUT on-the-fly. Such symbols, along with any symbol transitions that did not occur in training are given a nominal probability in the state transition matrix ($P = 0.05$). This is done to avoid otherwise correct chains being assigned zero probabilities.

Then, the probability of a model matching the observation sequence is calculated as $P(w|s) = \pi \prod_{t=1}^{l} P_w(s_t|s_{t-1})$, where l is the length of the word in the test sequence and π is the prior probability of a chain starting in any one of its states, here set to $\pi = 1$. For the isolated sign (i.e. non continuous) experiments reported in this paper, we use manually segmented sign sequences from which we obtain l.

For continuous recognition, a Viterbi algorithm can be used to choose the state sequence, and hence sign, that maximises the overall probability whilst simultaneously estimating the sign boundaries.

5 Performance Results

The lexicon was selected by randomly choosing 164 common signs. Unlike previous approaches, signs were not selected to be visually distinct but merely to represent a suitable cross section of signs from the BSL level 1 vocabulary.

Video sequences were collected for a single person performing each of the 164 signs with 10 repetitions of each sign totalling 2 hours of video footage (1640 individual signs). The video was hand labelled for groundtruth. The data is divided into training and test sets by random selection. The proportions of test and training data was then varied over the classification experiments.

A classifier bank consisting of 164 Markov chains, one for each individual sign was learnt from training data. The unseen test data was then presented to the classifier bank and the most probable word determined using the approach described in Section 4.3. Table 3 demonstrates the classification performance for both the inclusion and exclusion of DEZ with varying number of training examples. In these experiments, if there are n training examples, then the classification is tested on the remaining $10 - n$ examples for

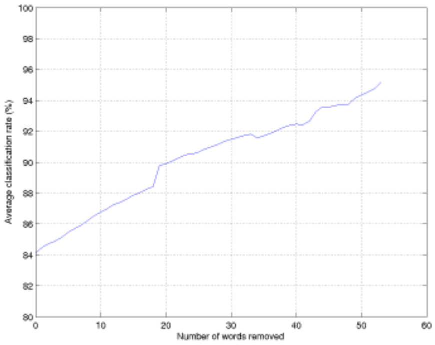


Figure 5: Average classification performance for 10 runs of lexical selection.

each sign. The results shows the average over 10 random selections of the training and test sets for each number of training examples.

As can be seen the inclusion of DEZ provides a 10% performance boost, reaching an average rate of around 89%. This is an impressive performance boost given that per frame hand classification rates themselves were only 75%. It is a clear indication that the DEZ features disambiguate many of the motion elements of signs within the lexicon. It is also impressive that such high classification rates can be achieved using only a single training instance. Of course increasing the number of training examples also increases the classification rate. However, this benefit quickly diminishes as the number of examples increases. This is evidence that although our stage I classification is not perfect it does however generalise well.

As previously stated when our lexicon was chosen, no attention was paid to ensuring signs were visually distinct, indeed without DEZ many signs form the same motion patters in stage I classification. It is the inclusion of DEZ that helps disambiguate many of the signs. The descriptors that we employ are merely a subset of those used in linguistic notation and the inclusion of additional descriptors will allow us to disambiguate further. However, sign ambiguity is inevitable as context and grammar are often used to provide a many to many mapping between signs and meaning. To identify signs which are ambiguous we separate a validation set from the training set. In the experiments below, the data is randomly divided into 5 training and 5 test sequences for each sign. Two of the training examples are used to learn the sign, and the most ambiguous signs are then identified by testing on the three remaining validation examples. (The most ambiguous signs are deemed to be the ones that are mis-classified). This process is repeated 10 times and the average results are shown in Figure 5. The curve shows the performance as the words identified as ambiguous on the validation set are progressively removed. It should be noted that these results can only be compared with those for N=2 in Table 3, since here we can only train on two examples. Removal of the 20 most ambiguous words results in a the performance improvement of some 6% – from 84% to 90%. It would not be unreasonable to expect a similar performance boost for the other cases of training number in Table 3 given sufficient data.

It is important to note that these high classification results have been obtained without constraints on grammar and can be compared to other viseme level approaches based upon HMM's where thousands of training examples were required to achieve just 22 words.

6 Conclusions

Our current demonstrator is capable of running at frame rate on a regular desktop PC. Due to the generalisation of features, and therefore the simplification in training, chains can be trained with new signs on the fly with immediate classification. This is something that would be difficult to achieve with traditional HMM approaches. However, the real power of this approach lies in its ability to produce high classification results on 'one shot' training and can demonstrate real time training on one individual with successful classification performed on a different individual performing the same signs. For much larger lexicons the representation may be augmented in various ways, for example through the use of internal DEZ features to capture information about the positions of the fingers in closed hand-shapes, or by allowing discrimination within hand-shape groups.

Acknowledgements

This work is funded by INRIA and the EC Project CogViSys.

References

[1] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *Proc. 8th European Conference on Computer Vision, Prague*, pages 391–401, 2004.

[2] D. B. (ed). *Dictionary of British Sign Language*. British Deaf Association UK, Faber and Faber, ISBN: 0571143466, 1992.

[3] Y. Fruend and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[4] R. Schapire. The boosting approach to machine learning: An overview. In *Proc. of MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

[5] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *Intl. Conf. on Automatic Face and Gesture Recognition*, pages 189–194, 1995.

[6] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *Proc. Intl. Conf. on Computer Vision*, volume II, pages 1063–1070, 2003.

[7] R. Sutton-Spence and B. Woll. *The Linguistics of British Sign Language, An Introduction*. Cambridge University Press, 1999.

[8] P. Viola and M. Jones. Robust real-time object detection. In *Proc. of IEEE Workshop on Statistical and Computational Theories of Vision*, 2001.

[9] C. Vogler and D. Metaxas. Asl recognition based on a coupling between hmms and 3d motion analysis. In *Proc. Intl. Conf. on Computer Vision*, pages 363–369, 1998.

[10] Z. Zhang, M. Li, S. Li, and H. Zhang. Multi-view face detection with floatboost. *Proc. of the Sixth IEEE Workshop on Applications of Computer Vision*, 2002.