Employing signed TV broadcasts for automated learning of British Sign Language

Patrick Buehler[1], **Mark Everingham**[2], **Andrew Zisserman**[1]

[1]Department of Engineering Science, University of Oxford, UK
[2]School of Computing, University of Leeds, UK
patrick@robots.ox.ac.uk

Abstract

We present several contributions towards automatic recognition of BSL signs from continuous signing video sequences: (i) automatic detection and tracking of the hands using a generative model of the image; (ii) automatic learning of signs from TV broadcasts of single signers, using only the supervisory information available from subtitles; (iii) discriminative signer-independent sign recognition using automatically extracted training data from a single signer. Our source material consists of many hours of video with continuous signing and aligned subtitles recorded from BBC digital television. This is very challenging material *visually* in detecting and tracking the signer for a number of reasons, including self-occlusions, self-shadowing, motion blur, and in particular the changing background; it is also a challenging *learning* situation since the supervision provided by the subtitles is both weak and noisy.

1 Introduction

The goal of this work is to automatically learn British Sign Language (BSL) signs from TV footage using the supervisory information available from subtitles broadcast simultaneously with the signing (see Figure 1). Previous research in sign language recognition has typically required manual training data to be generated for the sign *e.g.* a signer performing each sign in controlled conditions – a time-consuming and expensive procedure.

The main idea is to use a given English word to select a set of subtitles which contain the word – these form the positive training set – and a much larger set of subtitles that do not contain the word – these form the negative set. The sign that corresponds to the English word is then found using a multiple instance learning approach. This is a tremendously challenging learning task given that the signing is continuous and there is certainly not a one to one mapping between signs and subtitle words.

In order to learn a sign we require that it is signed several (more than 5) times by a single signer within one broadcast. However, we show that by adding an additional discriminative training phase, we are able to recognize this sign when signed by new signers within a restricted temporal search region.

Previous work on automatic sign extraction has considered the problem of aligning an American Sign Language sign with an English text subtitle, but under much stronger supervisory conditions (Farhadi and Forsyth, 2006; Nayak et al., 2009). Cooper and Bowden (2009) aim to automatically learn signs using the a-priori data mining algorithm, although without hand shape cues.

Outline. Knowledge of the hand position and hand shape is a pre-requisite for automatic sign language recognition. Section 2 presents our method for hand detection and tracking which uses a generative model of the image, accounting for the positions and self-occlusions of the arms. The results using this method exceed the state-of-the-art for the length and stability of continuous limb tracking.

Figure 1: **Example results.** The signs for "golf" and "tree" performed by two different signers are learned automatically. Our data is TV footage with simultaneously broadcast subtitles. Using an upper body pose estimator (Section 2), we find the location of the hands and arms in all frames. Knowing the hand position in each frame, signs are automatically learned from TV footage using the supervisory information available from subtitles (Section 3). With this method, a large number of signing examples can be extracted automatically, and used to learn discriminative sign classifiers (Section 4).

Section 3 describes our method for learning the translation of English *words* to British Sign Language *signs* from many hours of video with simultaneous signing and subtitles (recorded from BBC digital television). A multiple instance learning framework is used to cope with the misalignment between subtitles and signing and noisy supervision. Using the method we can learn over 100 signs completely automatically.

Lastly, Section 4 shows how the automatic recognition of signs can be extended to multiple signers. Using automatically extracted examples from a single signer we train discriminative classifiers and show that these can successfully recognize signs for unseen signers.

2 Hand and arm detection

In this section we describe our method for locating a signer's hands in the video. Previous approaches to hand tracking have applied skin colour models (Cooper and Bowden, 2007; Holden et al., 2005; Farhadi et al., 2007;

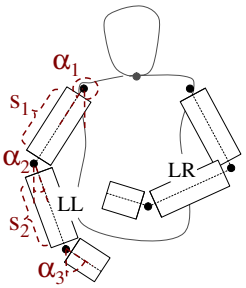


Figure 2: **Upper body model.** The pose is specified by 11 parameters – 5 for each arm and an additional binary parameter b indicating which arm is closer to the camera and hence visible in the case that the arms overlap. The shape of the head and torso and position of the shoulders are estimated in a pre-processing stage separate to estimation of the arm configuration.

Starner et al., 1998) or sliding window hand detectors (Kadir et al., 2004). These methods perform poorly when the hands overlap or are in front of the head, and lose track due to the ambiguities that routinely arise, resulting in poor estimation of hand position or unreliable assignment of hands to left or right. In contrast, by using a full upper body model (Figure 2) and accounting for self-occlusion our method proves capable of robust tracking for long videos, *e.g.* an hour, despite the complex and continuously changing background (the signer is overlaid on the TV programme). Figure 5 shows example output of the tracker.

The remainder of this section outlines our upper body pose estimator which tracks the head, torso, arms and hands of the signer; further details can be found in Buehler et al. (2008). In the following, we refer to the arm on the left side of the image as the "left" arm, and respectively the arm on the right side of the image as the "right" arm.

2.1 Approach

Estimation of the signer's pose is cast as inference in a graphical model of the upper body. To reduce the complexity of modelling and inference, the pose estimation process is divided into two stages (see Figure 3): (i) the shape of the head and torso and the position of the shoulders are estimated using a 2-part pictorial structure. This is relatively straightforward, and is described in Buehler et al. (2008); subsequently, (ii) the configuration of both arms and hands are estimated as those with maximum probability given the head and torso segmentations.

Generative model. Formally, given a rectangular sub-image $\mathbf{I}$ that contains the upper body of the person and background, we want to find the arm and hand configuration $\mathbf{L} = (b, \mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_n)$ which best explains the image, where $\{\mathbf{l}_i\}$ specifies the parts (limbs) and b is a binary variable indicating the depth ordering of the two arms. In our application we deal with $n = 6$ parts: the left and right upper arms, the lower arms and the hands. The appearance (*e.g.* colour) and shape of the parts are learned from manual annotation of a small number of training images. The background is continuously varying, and largely unknown.

Every part $\mathbf{l}_i = (s_i, \alpha_i)$ is specified by two parameters: scale (*i.e.* length of a part modelling foreshortening) s_i and rotation α_i, and by the part to which it is connected. The connections are in the form of a kinematic chain for the left and right arm respectively (see Figure 2).

We define the probability of a given configuration $\mathbf{L}$ conditioned on the image $\mathbf{I}$ to be

$$p(\mathbf{L}|\mathbf{I}) \propto p(\mathbf{L}) \prod_{i=1}^{N} p(\mathbf{c}_i|\lambda_i) \prod_{j\in\{LL,LR\}} p(\mathbf{h}_j|\mathbf{l}_j) \quad (1)$$

where N is the number of pixels in the input image, $\mathbf{c}_i$ is the colour of pixel i, and $\mathbf{h}_j$ is a HOG descriptor computed for limb j (see below).

The formulation incorporates two appearance terms (described in more detail below) modelling the agreement between the image $\mathbf{I}$ and configuration $\mathbf{L}$. The first, $p(\mathbf{c}_i|\lambda_i)$, models the likelihood of the observed pixel colours. Given the configuration $\mathbf{L}$, every pixel of the image is assigned a label $\lambda_i = \Lambda(\mathbf{L}, b, i)$ which selects which part of the model is to explain that pixel (background, torso, arm, etc.). The depth ordering of the two arms is given by the binary variable b which specifies which arm is closer to the camera and hence visible in the case that the arms overlap. The "labelling" function $\Lambda(\mathbf{L}, b, i)$ is defined algorithmically essentially by rendering the model (Figure 2) in back-to-front depth order (the "painter's algorithm") such that occlusions are handled correctly. For a given pixel, the colour likelihood is defined according to the corresponding label. Note that the pixel-wise appearance term in Eqn. 1 is defined over *all* pixels of the image, including background pixels not lying under any part of the model.

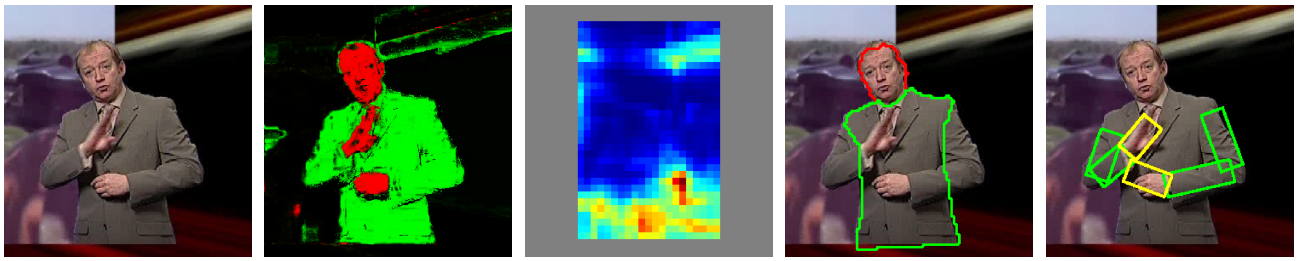
The second appearance term, $p(\mathbf{h}_j|\mathbf{l}_j)$, models the likelihood of observed gradients in the image (Figure 3c). This is based on Histogram of Oriented Gradient (HOG) (Dalal and Triggs, 2005) templates for the left and right lower arms, learned individually for different angles and scales. The HOG descriptor captures local information about image edges and shading with a controlled degree of photometric and spatial invariance. By using these descriptors, we exploit both boundary and internal features to determine the position and configuration of a limb.

The third term, $p(\mathbf{L})$, models the prior probability of configuration $\mathbf{L}$. This places plausible limits on the joint angles of the hands relative to the lower arms, and enforces the kinematic chain.

Complexity of inference. There are 11 degrees of freedom in the model: 5 for each arm and 1 for the depth ordering. The state spaces of the arm parts are discretised into 12 scales and 36 orientations. The hand orientation is restricted to be within 50 degrees relative to the lower arm and discretised into 11 orientations. Hence, the total number of possible arm configurations is $2 \times ((12 \times 36)^2 \times 11)^2 \approx 10^{13}$. Brute force optimisation over such a large parameter space is not feasible – the method described in the next section addresses this problem.

2.2 Computationally Efficient Model Fitting

The vast number of possible limb configurations makes exhaustive search for a global minimum of the complete



(a) Input image (b) Colour term (c) Gradient term (d) Torso/head shape (e) Arm pose

Figure 3: **Overview of pose estimation process.** Pose estimation for a given image (a) is performed using colour-based likelihoods (b) and likelihoods based on image gradients (c). The colour term in (b) is visualised by assigning the posterior probability for skin and torso to red and green colour channels respectively. The visualisation of the gradient term in (c) shows, for a given HOG template with fixed orientation and foreshortening, the likelihood at all locations in the image, where red indicates high likelihood. The example shown is for the right lower arm with rotation and foreshortening set to the ground truth values. Note the maximum is at the true centre point of the right lower arm in the image. Using the colour term (b) the head and torso can be segmented (d). The arm pose (e) is then estimated using the estimated torso and head shape, and both colour and gradient terms.

cost function infeasible. We therefore propose a fast approach based on a *stochastic* search for each arm, using an efficient sampling method (Felzenszwalb and Huttenlocher, 2005) to propose likely candidate configurations. Tree-structured pictorial structures are well suited for this task since samples can be drawn efficiently from this distribution (Felzenszwalb and Huttenlocher, 2005). However, they have several shortcomings explained in Buehler et al. (2008), *e.g.* the over-counting of image evidence. We show that by *combining* a sampling framework to hypothesise configurations with our full modelling of occlusion and background to assess the quality of the sampled configurations, we obtain the robustness of our complete generative model with the computational efficiency of tree-structured pictorial structure models.

The posterior distribution from which samples are drawn is given in Felzenszwalb and Huttenlocher (2005) as

$$p\left(\mathbf{L}|\mathbf{I}\right) \propto p\left(\mathbf{L}\right) \prod_{i=1}^{n} p\left(\mathbf{C}_i|\mathbf{l}_i\right) \qquad (2)$$

where $\mathbf{L} = (\mathbf{l}_1, ..., \mathbf{l}_n)$ defines the configuration of each part and $\mathbf{C}_i$ refers to the pixels covered by part i. $p(\mathbf{L})$ is defined as in Section 2.1 and places plausible limits on the joint angles of the hands relative to the lower arms.

The appearance term, $p\left(\mathbf{C}_i|\mathbf{l}_i\right)$, is composed of the product of pixel likelihoods using colour distributions modelled by mixtures of Gaussians, and edge and illumination cues added through HOG descriptors.

Sampling from Eqn. 2 is facilitated by the restriction to tree-like topologies and can as a result be performed in time linear in the number and configurations of parts (Felzenszwalb and Huttenlocher, 2005).

Improvements in sampling efficiency. When using a sampling method to propose plausible arm locations, it is important that the true arm configuration is contained in the set of samples. In this respect the tree-structured pictorial structure sampler is insufficient; for example, given an image where a part is partially or completely occluded, the associated probability for this part to be generated from its true location can be very low. To increase the probability of sampling the true configuration, we propose several modifications in Buehler et al. (2008), such as sampling from the max-marginal instead of the marginal distribution which is typically used.

Figure 4: **Example training data for the target sign 'tree'**. The top three rows are positive subtitle frame sequences (each around 20 seconds long), selected because they contain the text word 'tree'. However, the sign only appears in the first two (outlined in yellow). The final row is an example negative subtitle sequence which does not contain the text word 'tree' and also does not, in fact, contain the sign for tree. Signs are learnt from such weakly aligned and noisy data.

3 Automatic sign learning

This section outlines our approach for automatically learning signs from signed TV broadcasts. We describe how *weak* supervision is extracted from subtitles, visual description and matching of signs, and a multiple instance learning method for learning a sign despite the weak and noisy supervision. A more detailed discussion of the method can be found in Buehler et al. (2009).

Figure 5: **Sample of tracking results on hour-long sequences.** The estimated pose is shown for uniformly spaced frames in three hour-long sequences with different signers. The pose is qualitatively highly accurate in all frames.

3.1 Automatic generation of training data

By processing subtitles we can obtain a set of video sequences labelled with respect to a given target English word as 'positive' (likely to contain the corresponding sign) or 'negative' (unlikely to contain the sign); this is illustrated in Figure 4. Hand detection using our articulated upper body tracker (Section 2), and feature extraction are then applied to extract visual descriptions for the sequences.

To reduce the problems of polysemy and visual variability for any given target word we generate training data from the same signer and from within the same topic (*e.g.* by using a single TV program). Even when working with the same signer, the intra-class variability of a given sign is typically high due to 'co-articulation' where the preceding or following signs affect the way the sign is performed, expression of degree (*e.g.* 'very') or different emotions, and varying locations relative to the body.

3.1.1 Text processing

Subtitle text is extracted from the recorded digital TV broadcasts by simple OCR methods (Everingham et al., 2006) (UK TV transmits subtitles as bitmaps rather than text). Each subtitle instance consists of a short text, and a start and end frame indicating when the subtitle is displayed. Typically a subtitle is displayed for around 100–150 frames.

Given a target *word* specified by the user, *e.g.* "golf", the subtitles are searched for the word and the video is divided into 'positive' and 'negative' sequences.

Positive sequences. A positive sequence is extracted for each occurrence of the target word in the subtitles. The alignment between subtitles and signing is generally quite imprecise because of latency of the signer (who is translating from the soundtrack) and differences in word/sign order, so some 'slack' is introduced in the sequence extraction. Consequently, positive sequences are, on average, around *400* frames in length. In contrast, a sign is typically around 7–13 frames long. This represents a significant correspondence problem.

The presence of the target *word* is not an infallible indicator that the corresponding *sign* is present – examples include polysemous words or relative pronouns *e.g.* signing "it" instead of "golf" when the latter has been previously signed. We measured empirically that in a set of 41 ground truth labelled signs only 67% (10 out of 15 on average) of the positive sequences actually contain the sign for the target word.

Negative sequences. Negative sequences are determined in a corresponding manner to positive sequences, by searching for subtitles where the target word *does not* appear. For any target word an hour of video yields around 80,000 negative frames which are collected into a single negative set.

The absence of the target *word* does not always imply that the corresponding *sign* is not present in the negative sequences. This is because different words might be signed similarly, or a sign might be present in the video but not appear in the subtitles (*e.g.* referred to as "it").

3.1.2 Visual processing

A description of the signer's actions for each frame in the video is extracted by tracking the hands via our upper body model (Section 2). Descriptors for the hand position and shape are collected over successive frames to form a *window* descriptor which forms the unit of classification for learning. The temporal length of the window is between 7 and 13 frames, and is learnt for each sign.

Hand shape description. The 'shape' of the hands is extracted by segmentation, and represented by a HOG descriptor (Dalal and Triggs, 2005; Kjellström et al., 2008). HOG descriptors are chosen for their ability to capture both boundary edges (hand silhouette) and internal texture (configuration of the fingers), and the contrast normalization they employ gives some invariance to lighting.

To deal with cases where the hands are overlapping or touching, descriptors for each hand and also for the pair of hands are extracted in parallel.

3.2 Measuring visual distance between signs

Our learning approach seeks temporal *windows* of video which represent the same sign, where a window is the concatenation of visual descriptors for a sequence of frames. In order to compare two such windows a distance function is needed which captures differences in position and motion of the hands and their appearance.

For each frame t of the window, each hand is described by a vector $\mathbf{x}(t) = \langle \mathbf{x}_{pos}, \mathbf{x}_{dez}, \mathbf{x}_{dez_P} \rangle$ which combines hand position (pos) and shape (dez) for both the individual hand and the combined hand pair (subscript P). The descriptor for a window $\mathbf{X}$ is the concatenation of the per-frame descriptors $\mathbf{x}(t)$.

In BSL one hand is dominant, while the position and appearance of the other hand is unimportant for some signs. We build this into our distance function. Given two windows $\mathbf{X}$ and $\mathbf{X}'$ the distance between them is defined as the weighted sum of distances for the right (dominant) and left (non-dominant) hands:

$$D(\mathbf{X}, \mathbf{X}') = d_R(\mathbf{X}, \mathbf{X}') + w_L d_L(\mathbf{X}, \mathbf{X}') \qquad (3)$$

where $d_L(\cdot)$ and $d_R(\cdot)$ select the descriptor components for the left and right hands respectively. The weight $w_L \leq 1$ enables down-weighting of the non-dominant hand for signs where it does not convey meaning. We refer to two windows $\mathbf{X}$ and $\mathbf{X}'$ as showing the same sign if their distance $D(\mathbf{X}, \mathbf{X}')$ is below a threshold τ. Section 3.3 describes how w_L and τ is learnt for each individual target sign.

The distance measure for the left and right hand alike is defined as a weighted sum over the distances of the position, shape and orientation components (we drop the hand subscript to simplify notation):

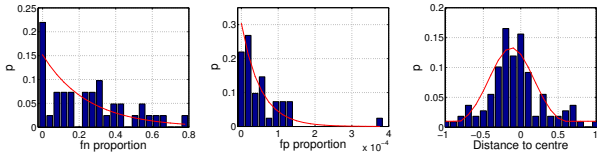
$$\begin{aligned} d(\mathbf{X}, \mathbf{X}') &= w_{pos} d_{pos}(\mathbf{X}, \mathbf{X}') + w_{dez} d_{dez}(\mathbf{X}, \mathbf{X}') \\ &\quad + w_{ori} d_{ori}(\mathbf{X}, \mathbf{X}') \end{aligned} \qquad (4)$$

The hand shape distance d_{dez} is computed with invariance to rotation. This is in accordance with linguistic sign research (Brien, 1993), where different hand configurations are described separately by shape (d_{dez}) and orientation (d_{ori}). The position distance d_{pos} is designed to be invariant to small differences in position, since repetitions of the same sign can be performed at different positions (*e.g.* this applies especially to signs performed in front of the chest). For a detailed description of these distance functions see Buehler et al. (2009).

The positive weights w_{pos}, w_{dez} and w_{ori} are learnt off-line from a small number of training examples.

3.3 Automatic sign extraction

Given a target word, our aim is to identify the corresponding sign. The key idea is to search the positive sequences to provide an example of the sign. Each positive sequence in turn is used as a 'driving sequence' where each temporal window of length n within the sequence is considered as a *template* for the sign. We require a score function to evaluate each template, with a high score if the template occurs within most of the positive sequences and not often in the negative data. The sign is determined by maximizing the score function over all templates, over the sign specific dominant/non-dominant hand weighting w_L, and over the threshold τ which indicates if two signing windows show the same sign..



(a) Positive S_+ (b) Negative S_- (c) Temporal prior S_t

Figure 6: **Distributions used to score template windows.** Plots (a) and (b) show the empirical distribution of errors (bars) and the fitted exponential distribution (curve). Note the scale on the x-axis. Plot (c) shows the temporal distribution of signs within corresponding positive sequences.

Multiple instance learning method. For a hypothesized setting of the classifier parameters $\theta = \{\hat{\mathbf{X}}, w_L, \tau\}$, with template window $\hat{\mathbf{X}}$ of length n, we assign a score

$$S(\theta) = S_+(\theta) + S_-(\theta) + S_t(\theta) \qquad (5)$$

to the classifier as a function of (i) its predictions on the positive sequences S_+ and the negative set S_-, and (ii) our prior knowledge about the likely temporal location of target signs S_t.

Unfortunately, when designing S_+ and S_-, we know that some non-negligible proportion of our 'ground truth' labels obtained via the subtitles will be incorrect, *e.g.* in a positive sequence the target word appears but the corresponding sign is not present, or in the negative data the target sign is present but not the corresponding target word. A model of such errors is empirically learned and approximated using exponential models (see Figure 6a,b).

The sign instances which correspond to a target word are more likely to be temporally located close to the centre of positive sequences than at the beginning or end. As shown in Figure 6c, a Gaussian model gives a good fit to the empirical distribution. The temporal prior p_t is learnt from a few training signs as for the score functions.

Searching for the sign by maximizing the score. Given a template window $\hat{\mathbf{X}}$ of length n from a positive sequence, the score function is maximized using a grid search over the weight for the left hand w_L, and over a set of similarity thresholds τ. This operation is repeated for all such template windows, with different lengths n, and the template window that maximizes the score is deemed to be the sign corresponding to the target word.

Using a per-sign window length allows for some signs being significantly longer than others. The weight w_L allows the importance of the left hand to be down-weighted for signs which are performed by the right hand alone.

3.4 Experiments

Given an English word our goal is to identify the corresponding sign. We deem the output a success if (i) the selected template window, *i.e.* the window with the highest score, shows the true sign (defined as a temporal overlap of at least 50% with ground truth) *and* (ii) at least 50% of all windows within the positive sequences which match the template window show the true sign.

Datasets. We tested our approach on 10.5 hours of signing sequences recorded from BBC broadcasts (including subtitles), aired between 2005 and 2006, and covering such diverse topics as politics, motoring and gardening. Signing is performed by three different persons. The image size after cropping the area around the signer is 300×330 pixels.

Test set. The method is evaluated on 210 words. These words were selected and fixed before running the experiments, without knowledge of the appearance of the target signs, *i.e.* how the corresponding sign is performed. Selection was based on: (i) the target word must occur more than 5 times in the subtitles; (ii) the target word is a verb, noun or adjective as opposed to linking words such as "then", "from", "the", etc.; (iii) the target word does not have multiple meanings (as opposed to *e.g.* the word "bank").
The full list of signs used is given at `www.robots.ox.ac.uk/~vgg/research/sign_language/`, which also contains example sequences of the detected signs.

Results. In 136 out of 210 cases (65%) we are able to automatically find the template window which corresponds to the target sign (see Figure 1 for two examples).
The precision-recall curve in Figure 7 (blue dashed line) shows that the score associated with a template window can be used as a confidence measure, giving an extremely good guide to success: at 11% recall (23 signs) precision is 100%; at a recall of 50% (105 signs) the precision is 77%.

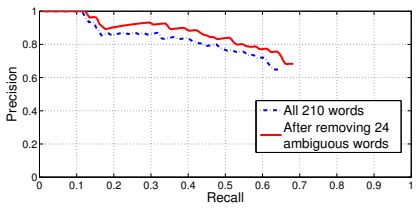


Figure 7: **Precision recall curve** computed using the score of the template window to rank learned signs.

Some words in our dataset co-occur with other words in the subtitles *e.g.* "prince" and "charles", which renders the correct template window ambiguous. Often these incorrectly-learned signs have a high associated score and hence reduce the precision even at low recall. By using simple statistics we can exclude 24 words from processing which leads to an improved precision-recall curve in Figure 7 (red solid line).
We achieve good results for a variety of signs: (i) signs where the movement of the hand is important *e.g.* "golf", (ii) signs where the hands do not move but the hand shape is important *e.g.* "animal"; (iii) signs where both hands are together to form a sign *e.g.* "plant"; (iv) signs which are finger spelled *e.g.* "bamboo"; (v) signs which are performed in front of the face *e.g.* "visitor", which makes identifying the hand shape difficult.
Some of the mistakes are understandable: For the word "wood", our result is the sign for "fire". This is not surprising since these two words often appeared together. The sign "year" is difficult since the signs for "last year", "this year" and "next year" differ – our method picks the sign for "next year".

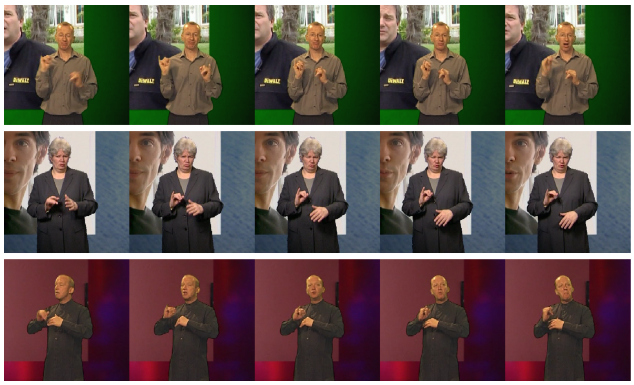


Figure 8: **Challenges for signer-independent sign recognition.** Repetitions of the same sign can differ in the hand position, the hand movement, the hand shape, and even in the number of hands involved. This is illustrated for three instances of the sign "bad".

4 Signer-independent sign recognition

Having demonstrated a method for learning a sign automatically, it is natural to investigate if the learnt sign can be recognized across different signers. The problem is that the learning method of Section 3 is built on the restrictions that apply to a single signer, for example that the lighting, body size and position do not vary significantly and also that (apart from co-articulations) the same sign is performed in a consistent style. These restrictions do not apply when the signer changes (see Figure 8) – signs can be performed with different speeds, variable extents, at varying locations, or with slightly different hand shapes. In many cases, these variations are due to the differences in signing between different people, such as local accents, or personal traits. The visual features and restrictions of Section 3 which took advantage of the single-signer situation are not sufficient for signer-independent recognition.
However, in this section we demonstrate that by adding a discriminative training stage signs *can* be recognized and localized in new signers. The experiments illustrate the extent to which our features (hand trajectory and hand shape) generalise to previously unseen signers.

4.1 Method

Our goal here is to *detect* signs in previously unseen signers using the automatically learnt signs from Section 3 within a temporally restricted search space. That is, instead of detecting a given sign in a full TV show (1 hour long), we search for it within short "positive sequences" extracted from around the word occurrences of the corresponding English word in the subtitles. In Section 3.3, a temporal prior is used to favour sign occurrences near the centre of a positive sequence. In this section, instead of such a prior, we use smaller positive sequences (on average half the length; 10 seconds long) extracted with a small offset from the word occurrence to take the empirically observed latency

between subtitles and signing into account (see Figure 6c). Note that empirically the sign we aim to detect is only performed in around half of the positive sequences, since the occurrence of a subtitle word does not imply the presence of the corresponding sign (see Section 3.1.1; although here, the positive sequences are shorter). Therefore, even a perfect sign detector would have an accuracy of at most 50%.

Assume that we have learnt a sign from a *training signer* using the method of Section 3. For a learnt sign we have an automatically learned template window $\hat{\mathbf{X}}$ with highest score, and all windows which are similar to the template, *i.e.* with a distance to $\hat{\mathbf{X}}$ below a threshold τ of the pairwise distance measure. We consider these windows as a positive training set.

We compare two methods for generalizing from the sign learnt from the training signer to recognizing this sign for other signers. (i) **Template matching:** The pairwise distance measure from Section 3.2 is used as classifier to identify signs which are similar to the learnt template window $\hat{\mathbf{X}}$. (ii) **Discriminative training:** A support vector machine (SVM) classifier is trained to detect the sign. Training data consists of the positive examples from the training signer, and negative examples taken from all the other signs considered.

For a given English word, a positive sequence for each word occurrence in the subtitles is extracted. Our aim is to find the corresponding sign in each of these sequences. This is achieved in a sliding window fashion by searching for the window with highest confidence according to (i) the corresponding SVM output or (ii) similarity to the learnt template. We search over different window lengths, since the duration of a sign in a positive sequence is unknown. In this way, one window is selected from each sequence.

Features. We use information from two cues: hand position and hand shape for each frame as described in Section 3. The hand shape cue is based on a set of hand exemplars which are used to describe the hand shape for each frame (think visual words); see Buehler et al. (2009) for a detailed description. The position of the left eye is automatically detected in each frame using the method of Everingham et al. (2006) and serves as reference point. Each training sample is down-sampled to be of equal temporal length (5 frames) – a prerequisite for SVM training.

SVM classifier. We use the LIBSVM library (Chang and Lin, 2001) to learn a binary SVM for each of the 15 different words in our dataset (see Section 4.2). Separate Radial Basis Function (RBF) kernels are computed for the hand position and the hand shape cue individually, and combined by computing the mean. We also evaluated using the product over the individual kernels instead, which gave comparable performance.

4.2 Experiments

Dataset. Experiments are performed for 15 English words, selected such that each word occurs more than five times in the subtitles of a specific signer (the training signer). The selected words are: better, Britain, car, help, hope, house, kitchen, money, mourn, new, night, room, start, team, and week.

For these 15 words, signing examples from the training signer are automatically extracted using our method from Section 3, and used to train initial SVM classifiers. Note that this includes wrongly learned signs, as is shown in Table 1, column "Learning - FP". These initial classifiers are subsequently used to extract additional signing examples from a database of 1730 positive sequences from 6 previously unseen signers (none of them being the training signer), each with a duration of 10 seconds. In this way, between 58 and 192 sequences are extracted for each word (Table 1, column "WS").

Results. First, we automatically learn for each of the 15 English words the corresponding sign. Even though the supervision provided by the subtitles is very weak and noisy, our results are highly accurate: out of 195 automatically extracted signing examples, 164 are correct (see Table 1, column "Learning"). Note that only the sign for "team" is not learned correctly (0 true positives TP, but 5 false positives FP).

From this dataset, an SVM classifier is learned for each word, and subsequently used to detect one example of the corresponding sign within each of the positive sequences. We define a ranking of the detections by confidence based on (i) the SVM decision value of the detected sign, and (ii) the margin between (i) and the second highest decision value within the same sequence (using non-maximum suppression). We know that the sign is only performed in about half the positive sequences (Section 4.1), hence assuming that our detector finds a sign a little less than half of the time (if it is performed), then the 20% highest ranked detections should often be correct. Indeed, for this subset, on average 67% of the detections show the true sign (see Table 1, column "SVM detector").

We further analysed the performance of our ranking function by plotting the proportion of correctly detected signs as a function of the highest ranked detections (Figure 9, blue curve).

The SVM classifiers used above were trained from automatically extracted signing examples, including 31 examples which do not show the correct sign (see Table 1, column "Learning – FP"). We observe a slight increase in accuracy if these examples are excluded from training (Figure 9, green solid curve).

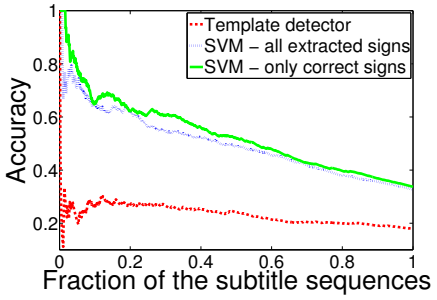


Figure 9: **Sign detection accuracy.** For each classifier the accuracy is shown as a function of detection confidence.

Comparison to template detector. We repeat the sign detection results from this section, using as a classifier the template-based distance function instead of the SVM approach. The results clearly deteriorate, as can be see in Figure 9 (red dotted curve) and in Table 1 (column "Template detector").

		Learning			SVM detector				Template detector		
Sign	**WS**	**TP**	**FP**	**Ratio**	**TP**	**FP**	**Ratio**	**SGR**	**TP**	**FP**	**Ratio**
better	95	11	2	0.85	2	2	0.50	2	2	15	0.12
Britain	98	9	1	0.90	18	13	0.58	3	1	17	0.06
car	63	15	1	0.94	6	4	0.60	4	15	4	0.79
help	150	22	1	0.96	6	3	0.67	3	15	20	0.43
hope	61	7	1	0.88	8	7	0.53	4	7	13	0.35
house	177	7	5	0.58	30	15	0.67	3	21	43	0.33
kitchen	58	7	0	1	6	0	1	3	2	24	0.08
money	102	6	1	0.86	10	3	0.77	3	7	15	0.32
mourn	77	8	0	1	10	2	0.83	2	4	11	0.27
new	183	20	4	0.83	47	7	0.87	4	2	13	0.13
night	62	8	0	1	11	0	1	3	3	4	0.43
room	126	9	0	1	10	0	1	2	2	2	0.50
start	192	17	10	0.63	26	67	0.28	5	5	5	0.50
team	151	0	5	0	0	4	0	2	0	41	0
week	135	18	0	1	20	6	0.77	2	6	23	0.21
MEAN	115			0.83			0.67	3.0			0.30

Table 1: **Recognizing signs in new signers.** For 15 English words, the corresponding signs are automatically learned (Section 3), and then used to recognize the sign in new signers (Section 4). Column "WS" shows the number of positive sequences for each word. For our automatic sign learning method, the number of correctly learned signs (TP), incorrectly learned signs (FP), and the ratio TP/(FP+TP) is given (column "Learning"). Subsequently, additional signing examples are detected within the 1730 positive sequences, either using the SVM framework as described in this section (column "SVM detector"), or the automatically found sign templates for each word (see Section 3) as detectors (column "template detector"). The number of new signers for which signing examples are extracted is given in column "SGR". Note that the values in the columns "SVM detector" and "Template detector" are computed using the 20% highest ranked sign detections (see also Figure 9, blue and red curves).

5 Conclusion

We described methods for visual tracking of a signer in complex TV footage, and for automatic learning of signs. The framework enables learning a large number of BSL signs from TV broadcasts using only supervision from the subtitles. We achieve very promising results even under these weak and noisy conditions.

We illustrated that examples automatically extracted for a single signer can be used to recognize a given sign for other signers provided an additional discriminative training stage is applied. This demonstrates that our features (hand trajectory and hand shape) generalise well across different signers, despite the significant inter-personal differences in signing.

Future work will concentrate on improving the accuracy of signer-independent recognition to a complete unconstrained scenario where no subtitles are available.

Acknowledgements. We are grateful for financial support from EPSRC, the Royal Academy of Engineering, ERC VisRec, and ONR MURI N00014-07-1-0182.

6 References

D. Brien. 1993. *Dictionary of British Sign Language*.

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

P. Buehler, M. Everingham, and A. Zisserman. 2009. Learning sign language by watching TV (using weakly aligned subtitles). In *Proc. CVPR*.

C. C. Chang and C. J. Lin. 2001. LIBSVM: a library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

H. Cooper and R. Bowden. 2007. Large lexicon detection of sign language. *IEEE Workshop on Human Computer Interaction*.

H. Cooper and R. Bowden. 2009. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *Proc. CVPR*.

N. Dalal and B Triggs. 2005. Histogram of oriented gradients for human detection. In *Proc. CVPR*.

M. Everingham, J. Sivic, and A. Zisserman. 2006. Hello! My name is... Buffy – automatic naming of characters in TV video. In *Proc. BMVC*.

A. Farhadi and D. Forsyth. 2006. Aligning ASL for statistical translation using a discriminative word model. In *Proc. CVPR*.

A. Farhadi, D. Forsyth, and R. White. 2007. Transfer learning in sign language. In *Proc. CVPR*.

P. Felzenszwalb and D. Huttenlocher. 2005. Pictorial structures for object recognition. *IJCV*, 61(1).

E. J. Holden, G. Lee, and R. Owens. 2005. Automatic recognition of colloquial Australian sign language. In *Workshop on Motion and Video Computing*.

T. Kadir, R. Bowden, E. J. Ong, and A. Zisserman. 2004. Minimal training, large lexicon, unconstrained sign language recognition. In *Proc. BMVC*.

H. Kjellström, J. Romero, D. Martínez, and D. Kragić. 2008. Simultaneous visual recognition of manipulation actions and manipulated objects. In *Proc. ECCV*.

S. Nayak, S. Sarkar, and B. Loeding. 2009. Automated extraction of signs from continuous sign language sentences using iterated conditional modes. In *Proc. CVPR*.

T. Starner, J. Weaver, and A. Pentland. 1998. Real-time American sign language recognition using desk- and wearable computer-based video. *IEEE PAMI*, 20(12).