Int J Comput Vis
DOI 10.1007/s11263-013-0672-6

Automatic and Efficient Human Pose Estimation for Sign Language Videos

James Charles · Tomas Pfister · Mark Everingham · Andrew Zisserman

Received: 4 February 2013 / Accepted: 29 October 2013

Abstract We present a fully automatic arm and hand tracker that detects joint positions over continuous sign language video sequences of more than an hour in length. To achieve this, we make contributions in four areas: (i) we show that the overlaid signer can be separated from the background TV broadcast using co-segmentation over all frames with a layered model; (ii) we show that joint positions (shoulders, elbows, wrists) can be predicted per-frame using a random forest regressor given only this segmentation and a colour model; (iii) we show that the random forest can be trained from an existing semi-automatic, but computationally expensive, tracker; and, (iv) introduce an evaluator to assess whether the predicted joint positions are correct for each frame. The method is applied to 20 signing footage videos with changing background, challenging imaging conditions, and for different signers. Our framework outperforms the state-of-the-art long term tracker by Buehler et al. (International Journal of Computer Vision 95:180–197, 2011), does not require the manual annotation of that work, and, after automatic initialisation, performs tracking in real-time. We also achieve superior joint localisation results to those obtained using the pose estimation method of Yang and Ramanan (Proceedings of the IEEE conference on computer vision and pattern recognition, 2011).

Keywords Sign language · Human pose estimation · Co-segmentation · Random forest

1 Introduction

A number of recent papers have demonstrated that signs can be recognised automatically from signed TV broadcasts (where an overlaid signer describes the broadcast) using only weak and noisy supervision (Buehler et al. 2009; Cooper and Bowden 2009; Farhadi and Forsyth 2006). For example, by using the correlations between subtitles and signs both Buehler et al. (2009) and Cooper and Bowden (2009) were able to automatically extract sign-video pairs from TV broadcasts; these automatically extracted sign-video pairs could then be used as supervisory material to train a sign language classifier Buehler et al. (2010) to recognise signs in new material. However, current research in this area has been held back by the difficulty of obtaining a sufficient amount of training video with the arms and hands of the signer annotated. This is a great pity because there is a practically limitless supply of such signed TV broadcasts.

The standard approach of Buehler et al. (2011) for tracking arms and hands in sign language TV broadcasts requires manual labelling of 64 frames per video, which is around three hours of manual user input per one hour of TV footage. In addition, the tracker (by detection) is based on expensive

Electronic supplementary material The online version of this article (doi:10.1007/s11263-013-0672-6) contains supplementary material, which is available to authorized users.

Mark Everingham, who died in 2012, made a significant contribution to this work. For this reason he is included as a posthumous author. An appreciation of his life and work can be found in Zisserman et al. (2012).

J. Charles · M. Everingham
School of Computing, University of Leeds, Leeds, UK
e-mail: j.charles@leeds.ac.uk

M. Everingham
e-mail: m.everingham@leeds.ac.uk

T. Pfister (✉) · A. Zisserman
Department of Engineering Science, University of Oxford, Oxford, UK
e-mail: tp@robots.ox.ac.uk

A. Zisserman
e-mail: az@robots.ox.ac.uk

Published online: 22 November 2013

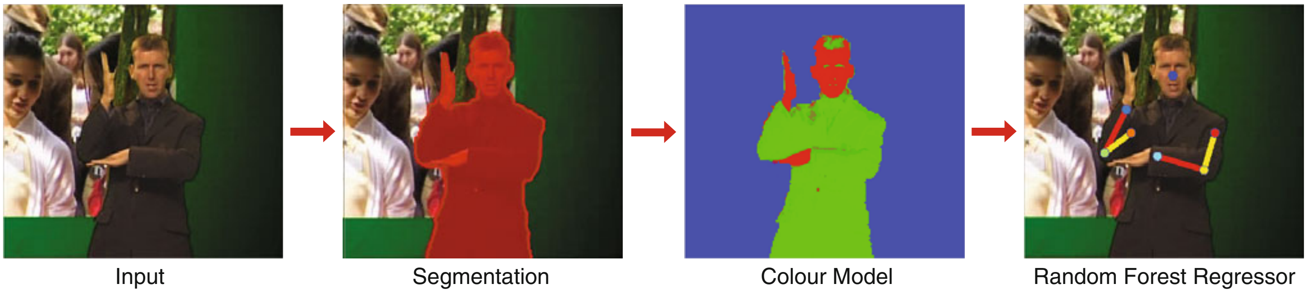


Fig. 1 Arm and hand joint positions are predicted by first segmenting the signer using a layered foreground/background model, and then feeding the segmentation together with a colour model into a random forest regressor

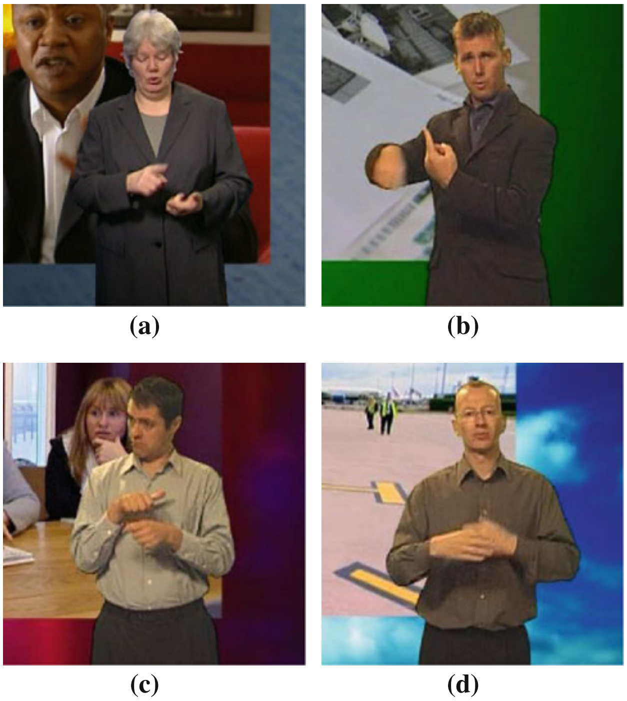


Fig. 2 Challenges for joint tracking. **a** Similar foreground and background colours render the colour cue less informative; **b** motion blur removes much of the edges of the arm; **c** a face in the background renders the face detector-based colour model initialisation difficult; **d** proximity of the two hands makes the assignment to left and right hand ambiguous

computational models and requires hundreds of seconds computation time per frame. Furthermore, signed TV broadcasts are very challenging material to segment and determine human joint positions on for a number of reasons that include: self-occlusion of the signer, self-shadowing, motion blur due to the speed of the signing, and, in particular, the changing background (since the signer is superimposed over a moving video that frequently even contains other people, e.g. see Figs. 1 and 2). These three factors have hindered the large scale application of this method.

In this paper we describe a method for tracking joint positions (of arms and hands) without any manual annotation and, once initialised, the system runs in real-time. The three key ideas are (i) for signed video the signer can be segmented automatically using co-segmentation (Sect. 2), (ii) given the segmentation, the joint positions can be predicted using a random forest, and (iii) the random forest can be trained using Buehler et al.'s tracking output, with no manual annotation (Sect. 3). We show that the random forest trained in this manner generalises to new signers (Sect. 5). Figure 1 illustrates the processing steps.

Each of the ideas has more general applicability: (i) the co-segmentation method can be easily generalised to other similarly laid out TV broadcasts, e.g. the majority of EU countries broadcast their signed TV broadcasts in a format suitable for this method; (ii) joint positions can be predicted by a random forest regressor in general, once the person is segmented from the background [as in the Kinect line of research Shotton et al. (2008)]; and (iii) the random forest tracker can be trained from existing tracked data with quite some generalisation over clothing and body mass (Charles et al. 2013).

This paper is an extended version of our BMVC 2012 paper Pfister et al. (2012). In addition to a more detailed exposition, we include here an extensive set of new experiments with a much larger dataset (20 TV broadcast videos instead of 5), a pose evaluator (Sect. 4) that provides an indication of whether the joint predictions are correct or not, and a comparison with the joint prediction method of (Yang and Ramanan 2011).

1.1 Related work

1.1.1 Random forests

The innate versatility of random forests (RFs) (Amit and Geman 1997; Breiman 2001) makes them suitable for a variety of machine learning tasks (Criminisi et al. 2012), such as classification, regression and clustering. They are naturally multi-class and contain a structure which lends itself to parallelisation and multi-core implementations (Sharp 2008). Along with these properties, the ever increasing computing power and training data over recent years has spurred the interest in RFs and fern-based Ozuysal et al. (2010) methods in computer vision literature. They have been applied to a variety of problems, including image classification (Bosch et al. 2007; Marée et al. 2005), object detection (Gall and Lempitsky 2009; Criminisi et al. 2011; Kontschieder et al. 2012), image/video segmentation tasks (Shotton et al. 2008; Yin et al. 2007; Geremia et al. 2011; Nowozin et al. 2011) and feature extraction (Liu et al. 2012). RFs are also fast to resolve at inference time, therefore lending themselves to real-time applications of tracking (Lepetit and Torr 2006; Santner et al. 2010; Apostoloff and Zisserman 2007).

In particular, we are interested in the work on human pose estimation where RFs have been used for head pose estimation (Fanelli et al. 2011) and detecting facial feature points (Fanelli et al. 2012; Dantone et al. 2012; Cootes et al. 2012). Of special regard are the methods for inferring full body pose, where notable success has been achieved using depth imagery. By applying a classification forest, Shotton et al. (2011) were able to segment a 3D depth map of a person into body parts and use the segmentation as a stepping stone for computing body joint locations. A performance boost was found by Girshick et al. (2011) using regression forests and Hough voting. Further improvements in accuracy on the same dataset were obtained by Taylor et al. (2012) using an RF to form dense correspondences between depth image pixels and a 3D body model surface, enabling the use of a one-shot optimisation procedure to infer pose. Recently Sun et al. (2012) have conditioned the RF on a global variable, such as torso orientation, to enhance performance.

The success of these full body pose estimation methods depends upon the use of depth imagery. Depth images are colour and texture-invariant and make background subtraction much easier. This substantially reduces the variability in human appearance. The remaining variability due to body shape, pose and camera angle is accounted for by training with large quantities of data. In the same spirit, we propose an upper body pose estimation method that exploits the large quantities of training data available and the efficiency and accuracy of RFs. However, our method does not depend upon depth imagery for success, but instead uses raw RGB images with only a partially known background.

1.1.2 Pose Estimation

There is a vast array of literature regarding human pose estimation due to a huge array of different applications reliant on analysing people in images and video (Moeslund 2011). It is common to use pictorial structures (Felzenszwalb and Huttenlocher 2005; Ramanan 2006; Ramanan et al. 2007; Sivic et al. 2006) to model human pose due to low computational complexity during inference. In more recent work, the focus has been on improving the appearance models used in pictorial structures for modelling the individual body parts (Eichner and Ferrari 2009; Eichner et al. 2012; Andriluka et al. 2012; Johnson and Everingham 2009; Sapp et al. 2010). Building upon the pictorial structure framework, Felzenszwalb et al. (2008, 2010) proposed deformable part based models. It has been shown by Yang and Ramanan (2011) that a mixture of deformable parts can be used in a tree structured model to efficiently model human pose. This results in a very general and powerful pose estimation framework which we compare to our method in Sect. 5.3.4. Sapp et al. (2011) model body joints rather than limbs, and also track joints across frames, using a set of tree-structured sub-models. We have not yet explored in our work the benefit of tracking the predicted joints over time.

Previous work on pose estimation for sign language recognition (Cooper and Bowden 2007; Starner et al. 1998a; Farhadi et al. 2007; Buehler et al. 2011; Pfister et al. 2012) in videos has relied on accurate hand tracking where it is popular to use skin colour for hand detection, although other detectors based on sliding window classifiers using Haar-like image features (Kadir et al. 2004; Ong and Bowden 2004; Dreuw et al. 2012) have been used. Of particular relevance here is the method of Buehler et al. (2011) which used a generative model for both the foreground (signer) and background (the image area surrounding the signer). The foreground was generated by rendering colour models of the limbs and torso in back-to-front depth order (the "painter's algorithm") so that occlusions were handled correctly. The computational expenses of evaluating all such renderings was reduced by sampling from a pictorial structure proposal distribution.

1.1.3 Co-segmentation

Co-segmentation methods (Rother et al. 2006; Hochbaum and Singh 2009; Joulin et al. 2010; Chai et al. 2011) consider sets of images where the appearance of foreground and/or background share some similarities, and exploit these similarities to obtain accurate foreground-background segmentations. Rother et al. (2006) originally introduced the problem of co-segmenting image pairs. Their approach was to minimise an energy function with an additional histogram matching term that forces foreground histograms of images to be similar. Hochbaum and Singh (2009) modified the histogram matching term to enable the use of max flow-based algorithms. More recently, Chai et al. (2011, 2012) proposed co-segmentation algorithms that work on each image category separately, and embed class-discriminative information into the co-segmentation process.

In our case our co-segmentation algorithm automatically separates signers from any signed TV broadcast by building a layered model (Jojic and Frey 2001; Szeliski et al. 2000; Kumar et al. 2008). We use this layered model of the signer to

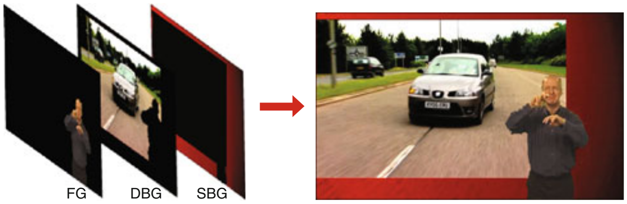


Fig. 3 Generative layered model of each frame. The co-segmentation algorithm separates the signer from any signed TV broadcast by building a layered model consisting of a foreground (FG), dynamic background (DBG) and static background (SBG)

provide a suitable input representation for the random forest regressor that is superior to using the raw input image itself.

1.1.4 Sign Language Recognition

Previous studies in sign language recognition rely on data generated by performers signing words under controlled conditions. Learning to recognise signs usually depends upon obtaining ground truth data and the ability to track the signers' head and hand positions (Vogler and Metaxas 1998; Dreuw et al. 2006; Starner et al. 1998b). Heavy constraints are typically imposed, such as wearing motion sensors (Chunli et al. 2002) or using a uniform background and/or wearing coloured gloves. Generating a small amount of such data with ground truth is both labour-intensive and expensive. It is possible to learn signs with small quantities of labelled data (Kadir et al. 2004; Bowden et al. 2004), but to increase the vocabulary of recognisable signs from 100s of words to 1,000s of words, more data is required. Methods exist which remove the need to annotate signs, and instead use weak and noisy supervision (Cooper and Bowden 2009; Buehler et al. 2009) from signed TV broadcasts. However, to release the full potential of these systems and harness the power of a larger dataset, one requires a fast and inexpensive method of tracking the signer. Here we show how to generate tracked signer data cheaply and in real-time.

2 Co-segmentation Algorithm

The goal of the co-segmentation algorithm is to segment the overlaid signer from each frame of the broadcast. We exploit the fact that sign language broadcasts consist of an explicit layered model as illustrated in Fig. 3. In the spirit of a generative model, i.e. one that generates the image by composition, we exploit these inherent layers to provide an accurate segmentation of the signer. We describe the three layers in the following paragraphs.

The static background layer (SBG) essentially consists of the framing (around the actual/original broadcast) that has been added by the studio. As can be seen in Fig. 4, the static background is partially revealed and partially occluded in each frame depending on the position of the signer. In a similar manner to how a "clean plate" is constructed in film post-production, by looking through the whole video and combining the partially revealed static backgrounds one can automatically, and almost fully, reconstruct the actual static background. This layer can then be exploited when segmenting the signer.

The dynamic background layer (DBG) consists of a fixed rectangle, where the original video is displayed, but is always partially covered by the signer and changes from one frame to another. Its colour information, for the region where it does not overlap a bounding box on the signer, is modelled separately and forms a background distribution for a subsequent segmentation of the signer.

Finally, the foreground layer (FG) consists of the moving signer. By assuming that the colour distribution of the signer remains constant we can build an accurate foreground colour model for the whole video.

2.1 Algorithm Overview

The input to the co-segmentation algorithm is a signed TV broadcast video, and the output is a foreground segmentation, a quality score for the segmentation, the head position and a colour model for the skin and torso. These will be used in the random forest regressor. The algorithm consists of two main steps:

2.1.1 Automatic Initialisation (Per Image Sequence)

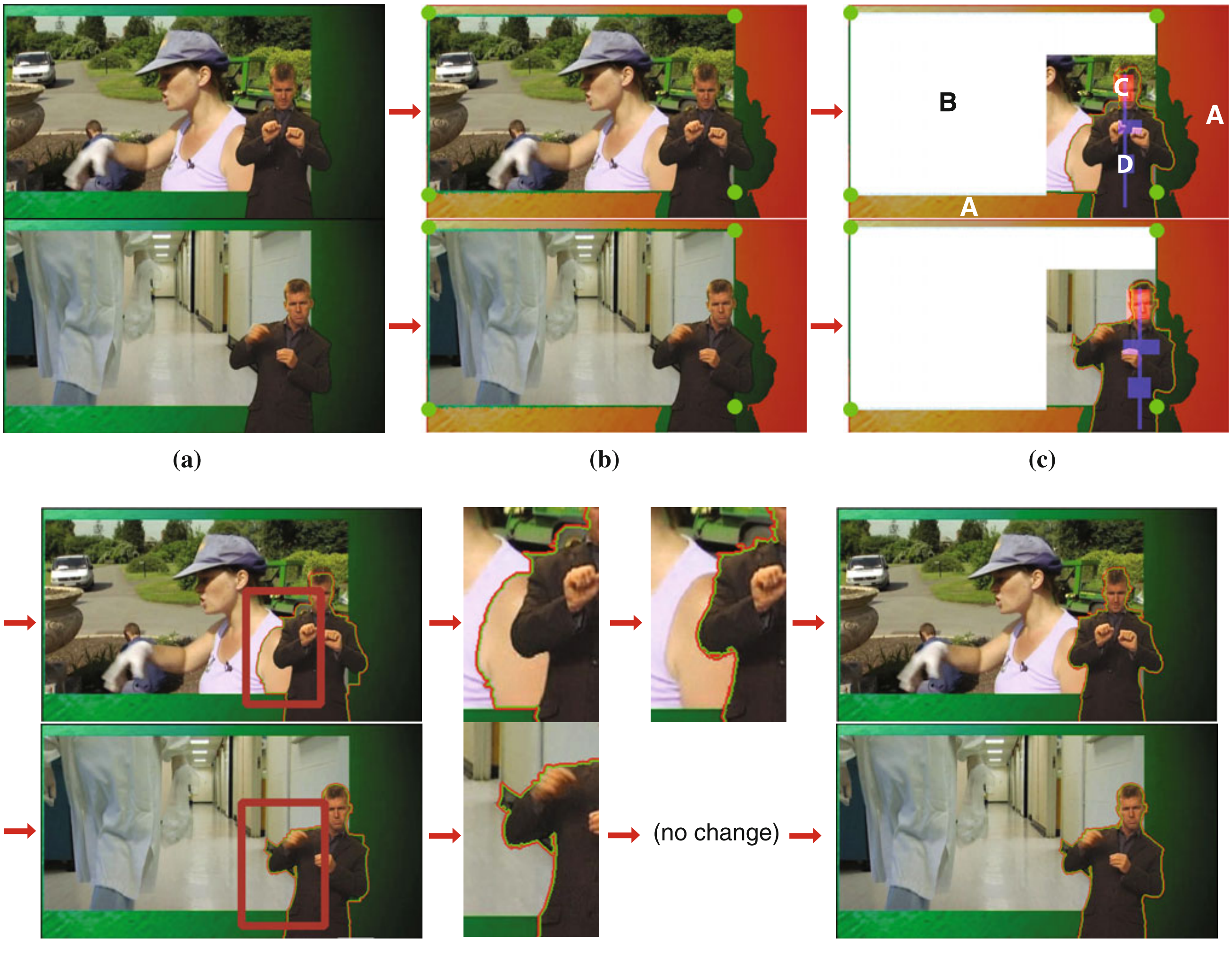
To exploit the inherent layered model we initialise the algorithm by first determining the "clean plate", the dynamic rectangle and the foreground colour model. The details of how this "initialisation set" is obtained are given in Sect. 2.2.

2.1.2 Segmentation with a Layered Model and Area Constraints (Per Frame)

The initialisation set is then used to derive an accurate hard segmentation of the signer in each frame. The clean plate and an area constraint are used to refine an initial rough segmentation. The details of this method are given in Sect. 2.3.

2.2 Co-segmentation Initialisation

Our goal here is to obtain the layers and their layout that are common to the video sequence (in order to enable the subsequent per-frame segmentation). In detail, we wish to obtain the regions shown in Fig. 4, as well as the foreground colour distribution. Our approach is to treat each frame as being generated from a number of layers, as depicted in Fig. 3, and to thereby solve for the layers and layout. This problem



(a) (b) (c)

(d) (e) (f)

Fig. 4 Co-segmentation. **a** Original frames; **b** dynamic layer (*rectangle* spanned by the *green dots*) and the permanently fixed background (in *red*)—the remaining green area behind the signer is the backdrop which is not part of the fixed background; **c** rough segmentation with clamping regions for running graph cut. *A* is the permanently fixed background; *B* is the clamping region for the dynamic background; *C* is part of the foreground colour model and *D* is a hard foreground clamp (based on the position of the detected face). **d** Initial GrabCut segmentation that uses colour distributions of *A*, *B* for background and *C*, *D* for foreground; **e** detail of the red rectangular region of (**d**) showing the segmentation refinement stage (see text); **f** segmentation after clean plate and area size refinements (Color figure online)

differs from typical applications of generative layered models for video, e.g. (Jojic and Frey 2001; Kumar et al. 2008), since part of the background in the video is always moving so we have a dynamic rather than fixed layer. The creation of the layered model can be broken down into a step per layer:

2.2.1 Dynamic Background

The aim in this step is to find the rectangle that contains the dynamic background, and furthermore divide it into a region where the signer may overlap, and another where the signer never reaches (see Fig. 4c). The latter region will be used to define a per-frame background colour. To this end we find pixels that change intensity values for the majority of frames and compute their rectangular bounding box, as shown in Fig. 4b. This also yields an area that is permanently static throughout the video (region A in the same figure) that we use as a permanent BG clamping region. Regions A and B in the same figure, which the signer never reaches, are defined relative to the position of the signer's face (the face detection method is described below).

2.2.2 Static Background

The aim here is to find the static background, which can be viewed as consisting of a "clean plate" (term explained above). Once we have this "clean plate", we can then say with near-certainty whether a pixel belongs to the FG or BG. The clean plate is obtained by roughly segmenting a random set of frames into FG (signer) and BG using a graph cut algorithm. The regions used to obtain the FG and BG distributions are illustrated in Fig. 4c. In particular, areas selected relative to the position of the signer's face (face detection method described below) are used to initialise the FG colour distribution. Given these segmentations, the clean plate is obtained as a median over the BG.

2.2.3 *Foreground Colour Model*

Here the aim is to obtain the signer colour distribution (which is assumed approximately constant throughout the sequence). This removes the need for finding accurate FG colour models for individual frames. The colour distribution (which is represented by a histogram) is obtained from the rough FG segmentations (Fig. 4c, computation described above) using frames where the colour histograms of the FG and dynamic background differ the most. The high colour difference increases the likelihood that there is a high contrast between the FG and BG and thus that the segmentation is correct.

2.2.4 *Face Detection*

Face detection is used for initialisation and frame-by-frame segmentation. Detection of both frontal and profile view faces is done by choosing between the face detector by Zhu and Ramanan (2012) (high recall for frontal faces) and a face detector based on upper body detection (Ferrari et al. 2008) (lower recall but detects profile views) according to their confidence values.

2.3 Per-frame Segmentation with a Layered Model and Area Constraints

Having finished the initialisation step we now have a layered model that can be used to derive a segmentation of the signer. This layered model (the "initialisation set") is used to (i) improve the segmentation by comparing each pixel against the clean plate (to yield a near-certain segmentation label as the background is known); and (ii) shrink the foreground segmentation size if it is too big (to avoid catching e.g. skin regions in the background).

The segmentation uses Rother et al. (2004), with the FG colour model provided by the initialisation set and, as in Ferrari et al. (2008), with the FG clamped in areas based on the face location (Fig. 4c). The BG colour distribution is known from the dynamic background. The segmentation is refined twice: first by comparing pixels to the clean plate of the static background, and then by shrinking the foreground size if it is much bigger than the average size. The latter is done by adding a constant to the graph cut unary potentials of the dynamic background (this increases the likelihood that a larger part of the dynamic background is labelled as BG, hence reducing the size of the FG). This addresses a common failure case where the dynamic background contains a colour similar to the signer, which leads to the foreground region 'catching' part of the dynamic background and becoming too large. In contrast, the foreground is seldom too small thanks to good FG colour model estimates. Examples of fully refined segmentations are shown in Fig. 4e.

The segmentation still fails in certain difficult cases, e.g. when the colours of the FG and BG are very similar or when the face detector fails. To this end we compute a segmentation quality score as described in Sect. 4.

2.4 Colour Model and Posterior

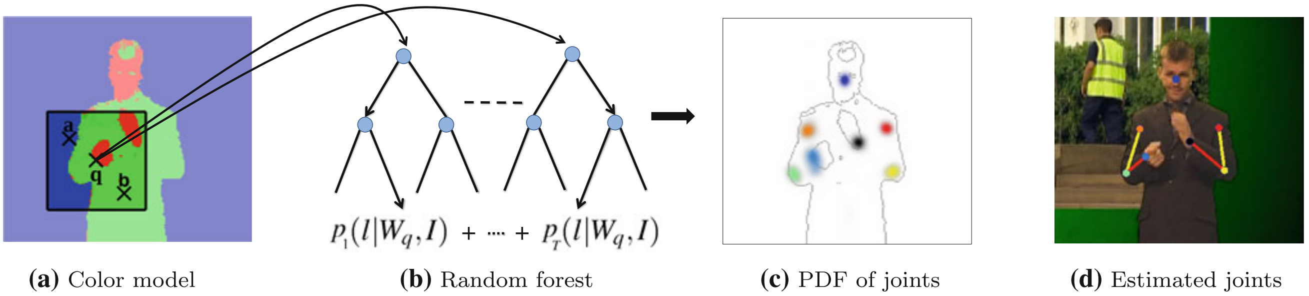
At this stage we have a foreground segmentation that is rated by a segmentation quality score. However, additional layout information is also available from the the spatial position of the the skin and torso (i.e. non-skin) pixels. The posterior probability of the skin and torso pixels is obtained from a colour model. Computing the colour *posteriors* for skin and torso abstracts away from the original colour, of the clothes for example, which varies between signers and is not directly informative (Benfold and Reid 2008).

In a similar manner to the construction of the initialisation set for the layers, the skin colour distribution is obtained from a patch of the face over several frames, and the torso colour distribution is obtained from a set of FG segmentations from which the colours of the face/skin are automatically removed. These colour distributions are then used to obtain a pixel-wise posterior for the skin and torso in each frame.

2.5 Technical Details

Here we provide the additional details for the segmentation method. The dynamic background is determined using a subset of 300 uniformly sampled frames for each video. Earth mover's distance (EMD) is used to compare colour histograms for extracting the foreground colour model and for generating colour posteriors (to remove skin regions from the FG segmentations). Faces are detected in the right half of the image for computational efficiency. The maximum foreground segmentation size is set to a standard deviation above the median segmentation size over all frames in a video.

The input to the random forest regressor (described in the following section) for each frame consists of: the foreground segmentation, the segmentation quality score, the head position, and the skin and torso posterior (from the colour model). The performance of the co-segmentation algorithm is assessed in Sect. 5.



(a) Color model **(b)** Random forest **(c)** PDF of joints **(d)** Estimated joints

Fig. 5 Estimating joint positions. **a** Input colour model image; **b** random forest classifies each pixel using a sliding window and learnt test functions; **c** probability density function of each joint location, shown in different colours per joint (more intense colour implies higher probability); **d** joint estimates, shown as small circles linked by a skeleton

3 Random Forest Regression

We cast the task of localising upper body arm joints and head position as a multi-class classification problem, classifying each image pixel into one of 8 categories $l \in$ {head centre, left/right wrist, left/right elbow, left/right shoulder, other} using a random forest classifier in a sliding-window fashion. From here on we also refer to "head centre" as a joint (see Fig. 5d). As shown in Fig. 5a, the input to the random forest comes from the colour model image after co-segmentation. The joints are localised on a per-frame basis to avoid tracking errors, e.g. drifting.

The random forest classifier uses simple features to make classification extremely computationally efficient. Classification to a discrete class label $l \in \{l_i\}$, for each pixel q across the image, is performed in a sliding-window fashion. We classify the pixels by computing the conditional distribution $p(l|W_q, I)$ for each label, where I is the colour model image and W_q is the set of pixels in the window surrounding q. The window size is chosen so as to maximise joint estimation accuracy in validation videos. The random forest is an ensemble of T decision trees, as illustrated in Fig. 5b. Each tree t consists of split nodes which perform a true or false test on incoming pixels. Pixels are recursively pushed down either the left or right branch depending upon the outcome of the test. When a pixel reaches a leaf at the bottom of the tree, a learnt probability distribution $p_t(l|W_q, I)$ assigns the pixel a probability for class label l. The final conditional distribution $p(l|W_q, I)$ is obtained by taking an average across all trees in the forest as follows:

$$p(l|W_q, I) = \frac{1}{T}\sum_{t=1}^{T} p_t(l|W_q, I) \qquad (1)$$

We use very efficient test functions $f(.)$ at the nodes of the trees which only compare pairs of pixel values (Shotton et al. 2008). A pixel q is represented by $\mathbf{x}_q = (x_q^1, x_q^2, x_q^3)$ where x_q^1, x_q^2, x_q^3 are the skin, torso and background colour posterior values at pixel q respectively (Benfold and Reid 2008). The function f operates on a pair of pixels (a, b) from within the window W_q and produces a scalar value which is compared against a threshold value υ—see Fig. 5a. These tests can take one of four forms: $f(a) = x_a^c$, or $f(a, b) = x_a^c - x_b^c$, or $f(a, b) = x_a^c + x_b^c$, or $f(a, b) = |x_a^c - x_b^c|$, where $c \in \{1, 2, 3\}$ indexes the type of colour posterior value to choose.

3.1 Training of the Forest

In each frame of the video, circular patches of radius 13 pixels centred on joint locations are labelled as that joint, with all other pixels labelled as 'other'. Each tree in the forest is trained by randomly sampling a diverse set of points S_n from the training frames. Each decision tree is trained recursively, with the split function and threshold at each node chosen to split the data reaching that node as "purely" as possible such that points belonging to the same class are sent to the same child node. The impurity of a split is measured using the Gini measure:

$$i(S_n) = 1 - \sum_{l} p(l|S_n)^2, \qquad (2)$$

where $p(l|S_n)$ is represented by a histogram of the dataset S_n over possible labels l at node n. The Gini impurity is chosen for its efficient implementation compared to e.g. information gain. We experimentally confirmed training time to be 1.5 times slower using information gain, with no significant difference in classification performance. Because there are many more 'other' pixels than 'joint' pixels, we balance the dataset by normalising the number of elements in the bin labelled l by the total number of elements in the training set labelled l. The parameters of split nodes are learnt by trying all possible test functions $f(.)$ and colour posterior types c for a randomly sampled offset pixel (a, b). The offset pixel is uniformly sampled within W_q, where $q \in S_n$. The data entering the node is split into a left subset S_n^L if $f(.) < \upsilon$ or otherwise to a right subset S_n^R.

The drop in impurity is measured as $\triangle i(S_n) = i(S_n) - P_L i(S_n^L) - (1 - P_L) i(S_n^R)$, where P_L is the fraction of data points that go to the left set. In each case the threshold value υ is chosen to maximise $\triangle i(S_n)$. The whole process is repeated k times (we use $k = 200$) and the set of parameters which maximise $\triangle i(S_n)$ overall is chosen as the winning decision function. This process is recursively repeated for all nodes. A node is declared a leaf node, and not split further, when (i) the maximum depth limit D of the tree has been reached or (ii) the node is pure i.e. all points reaching the node have the same class label. A per-leaf probability distribution $p_t(l|W_q)$ is stored at the leaf node, represented as a normalised histogram over the labels of all data points reaching the node.

3.2 Assigning Joint Locations

A location for the joint l is found by using the output of the random forest $p(l|W_q)$ and estimating the density of joint proposals using a parzen-window kernel density estimator with a Gaussian kernel. The position of maximum density is used as the joint estimate.

See Fig. 20 for an illustration of this method and comparison against ground truth.

4 Pose Evaluator

At this point our joint predictor outputs joint estimates for each frame of the video. However, the predictions are provided "as is", without an indication of whether they are correct or not. Therefore, in the spirit of Jammalamadaka et al. (2012) we train an evaluator that indicates whether a pose is correct or not. We accomplish this by analysing the failure cases and developing scores for predicting when the failures occur.

As pointed out in the introduction, we are blessed with a near-infinite amount of sign language interpreted TV broadcasts. Therefore, if necessary, frames for which pose estimates fail could be discarded with little loss. Detecting failures is hence particularly useful in our application, as with a fully functioning evaluator we could obtain near-perfect pose estimates for large parts of our videos. These joint estimates can then, in turn, be used to obtain accurate sign-video pairs for training a supervised sign language classifier (Buehler et al. 2010). From the perspective of the next stage in our pipeline [automatically extracting signs (Pfister et al. 2013)] where the pose estimation results will be used, the fact that the pose estimates for certain signs will be consistently incorrect, and therefore discarded by the evaluator, is very helpful, as we do not want to attempt to extract signs with incorrect pose estimates.

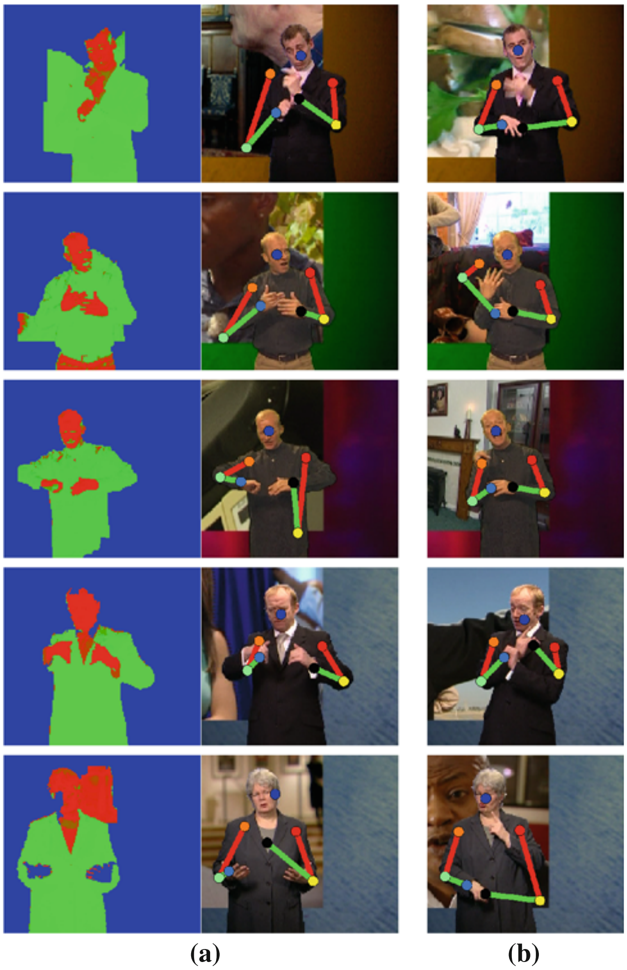


Fig. 6 Typical pose estimation errors. **a** Frames with segmentation failures, with the failed segmentation (*left*) and failed pose estimate (*middle*). **b** Frames where the left and right hands are confused. Poses estimates are illustrated with a colour coded skeleton

Figure 6 shows the main causes of failure: frames where the segmentation is faulty (≈ 80 % of errors), and where the left and right hand are confused (≈ 5 % of errors). The approach here will be to develop separate methods for detecting each of these failures. An SVM is trained to predict failed frames using the output of these methods as a feature vector. The classifier yields a simple lightweight evaluator that predicts whether the pose is correct or incorrect. The features for the classifier are discussed in Sects. 4.1 and 4.2, and details on the SVM that combines the features are given in Sect. 4.3.

4.1 Feature 1: Segmentation Score

The segmentations are generally fairly robust. However, occasionally they either oversegment or undersegment the foreground due to a similar foreground and background or due to face detection failures. This in turn results in wrong joint assignments.

One obvious way to detect failures is to compare the segmentations to ground truth segmentation masks. However, this would require significant manual labelling work which our automated joint detector was designed to avoid in the first place. Instead, we exploit our joint estimates by rendering a partial silhouette (Fig. 7a). This is done by rendering a rectangular binary mask for each limb given joint locations. Rectangles covering the head and arms are added according to the joint positions, and a rectangle covering the torso is added based on the shoulder positions. The partial silhouette can then be compared to the segmentation from the co-segmentation algorithm as shown in Fig. 7b, resulting in scores such as those in Fig. 8.

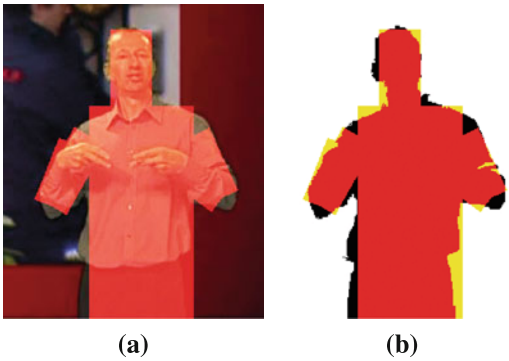


Fig. 7 Segmentation score for evaluator. **a** The silhouette (*red boxes*) rendered based on estimated joint positions. **b** The segmentation (*black*), rendered silhouette (*yellow*) and their overlap (*red*) which is used as a segmentation score (Color figure online)

Several segmentation scores are computed based on the output of this rendering. First, we compute a standard overlap score $o = \frac{T \bigcap A}{T \bigcup A}$ for comparing the two silhouettes, where T is rendered partial silhouette and A is the mask generated by the co-segmentation algorithm (Fig. 9). Second, a Chamfer distance between the silhouettes is also computed, yielding a measure of the similarity of the shapes of the silhouettes. Third, statistics based on the size of the segmentation are computed. These include absolute mask size $\|A\|$, difference between mask size and median mask size over all frames $\|M\|$: $\Delta = \frac{\|A\|-\|M\|}{\|M\|}$, Δ re-computed with temporally local medians, and differences between different Δ's. These scores form the first part of the feature vector for the evaluator classifier.

4.2 Feature 2: Mixed Hands

Another common error case is when the left and right hand are confused with each other, i.e. the left hand is connected to the right elbow and/or vice versa. In order to catch these failures we train a classifier with local histogram of oriented gradients (HOG) Dalal and Triggs (2005) features to detect correct and incorrect assignments. The tracking output from Buehler et al. (2011) is used as manual ground truth. The examples are clustered with K-means according to the hand-elbow angle and hand position into 15 clusters. One SVM is trained for

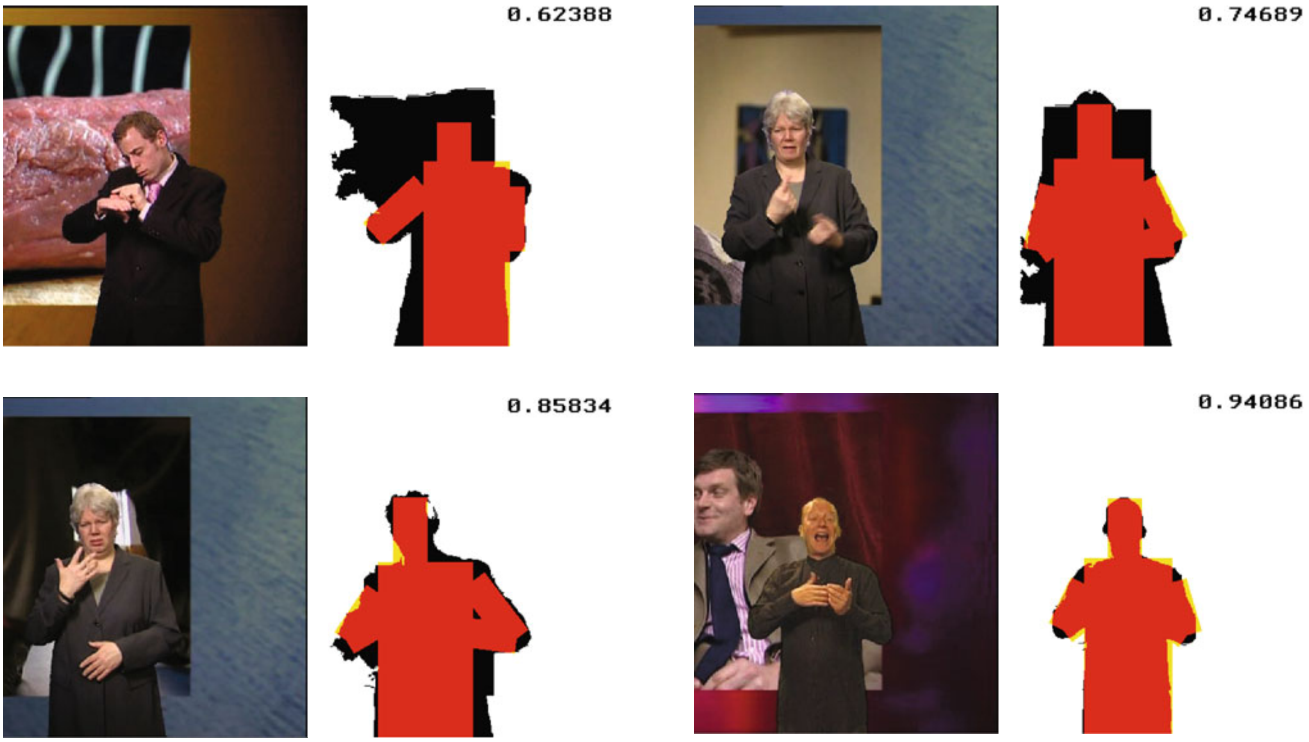


Fig. 8 Examples of frames with different segmentation overlap scores. The masks show the segmentation (*black*), rendered silhouette (*yellow*) and their intersection (*red*) (Color figure online)

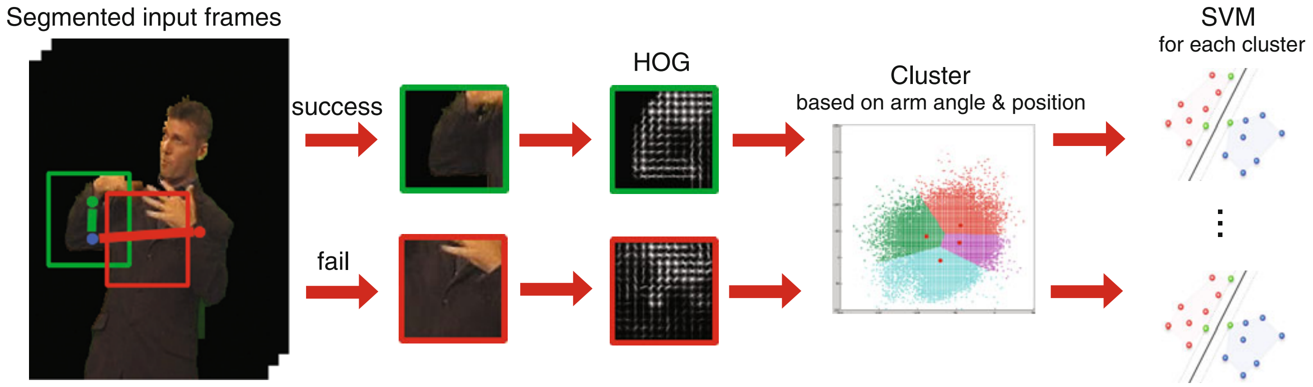


Fig. 9 Training the hand mixup detector. The evaluator is trained on HOG feature vectors placed in the middle of the correct and incorrect positions of the lower arm. Feature vectors are clustered into separate SVMs based on the hand-elbow angle and hand position

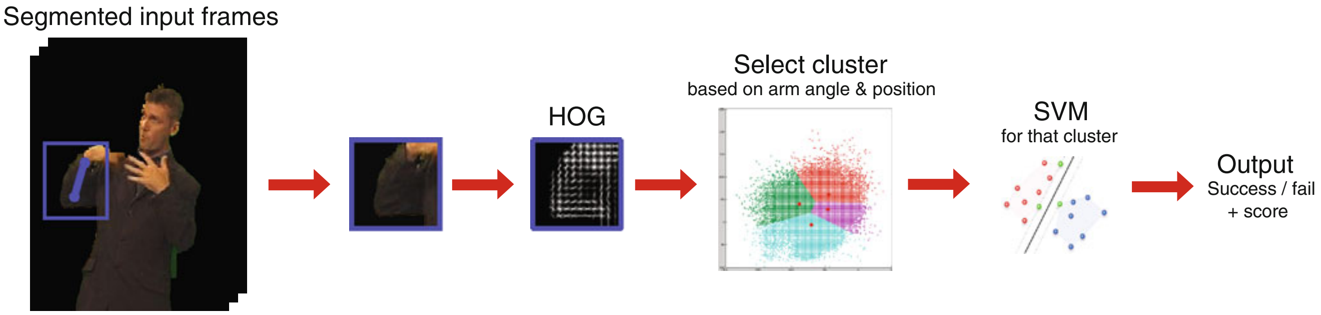


Fig. 10 Testing the hand mixup detector. The SVM trained on a cluster whose centroid best represents the predicted joints is chosen to evaluate the HOG feature vector placed in the middle of the hand and elbow positions. This SVM outputs a failure score which the evaluator exploits as a feature for predicting whether the pose estimate is successful or failed

each cluster as shown in Fig. 9. The HOG is computed in the middle of the lower arm. At test time, as shown in Fig. 10, predicted joints are assigned to the nearest cluster centroid based on hand-elbow angle and hand position. The SVM for this cluster is evaluated and the output score forms the second part of the feature vector for the evaluator classifier.

4.3 Evaluator: Uses the Above Features

The above features are then used to train an evaluator, which classifies the body pose estimate of each frame as either success or failure. Once the evaluator has been trained, at testing time frames classified as failures are discarded. Section 5.3 provides results without discarding frames, and Sect. 5.4 provides results with failed frames discarded.

To this end we train an SVM with a Chi-squared kernel based on the the above two feature sets (9 scores for segmentation—1 overlap score, 1 Chamfer score and 7 size statistics; and 1 score from the mixed hand classifier). An increase in accuracy was observed after adding each feature. The joint tracking output from Buehler et al. (2011) is used to automatically label the training set. This yields a simple lightweight evaluator (with a feature vector of dimension 10) that predicts whether the pose is correct or incorrect.

5 Experimental Results

First an overview of the dataset and evaluation criteria is presented (Sect. 5.1); then the performance of the co-segmentation algorithm, joint position estimator and pose evaluator are assessed (Sects. 5.2–5.4), and finally the computation time of the methods is discussed (Sect. 5.5). Sample videos demonstrating the methods, and a subset of the training data and annotations, are available online.[1]

5.1 Dataset and Evaluation Measure

Our dataset consists of 20 TV broadcast videos, each of which is between half an hour to one and a half hours in length. Each video typically contains over 40K frames of sign-interpreted video content from a variety of TV programmes. All frames of the videos have been automatically assigned joint labels using a slow but reliable tracker by Buehler et al. An example frame from each of the videos is shown in Fig. 11.

[1] http://www.robots.ox.ac.uk/~vgg/research/sign_language

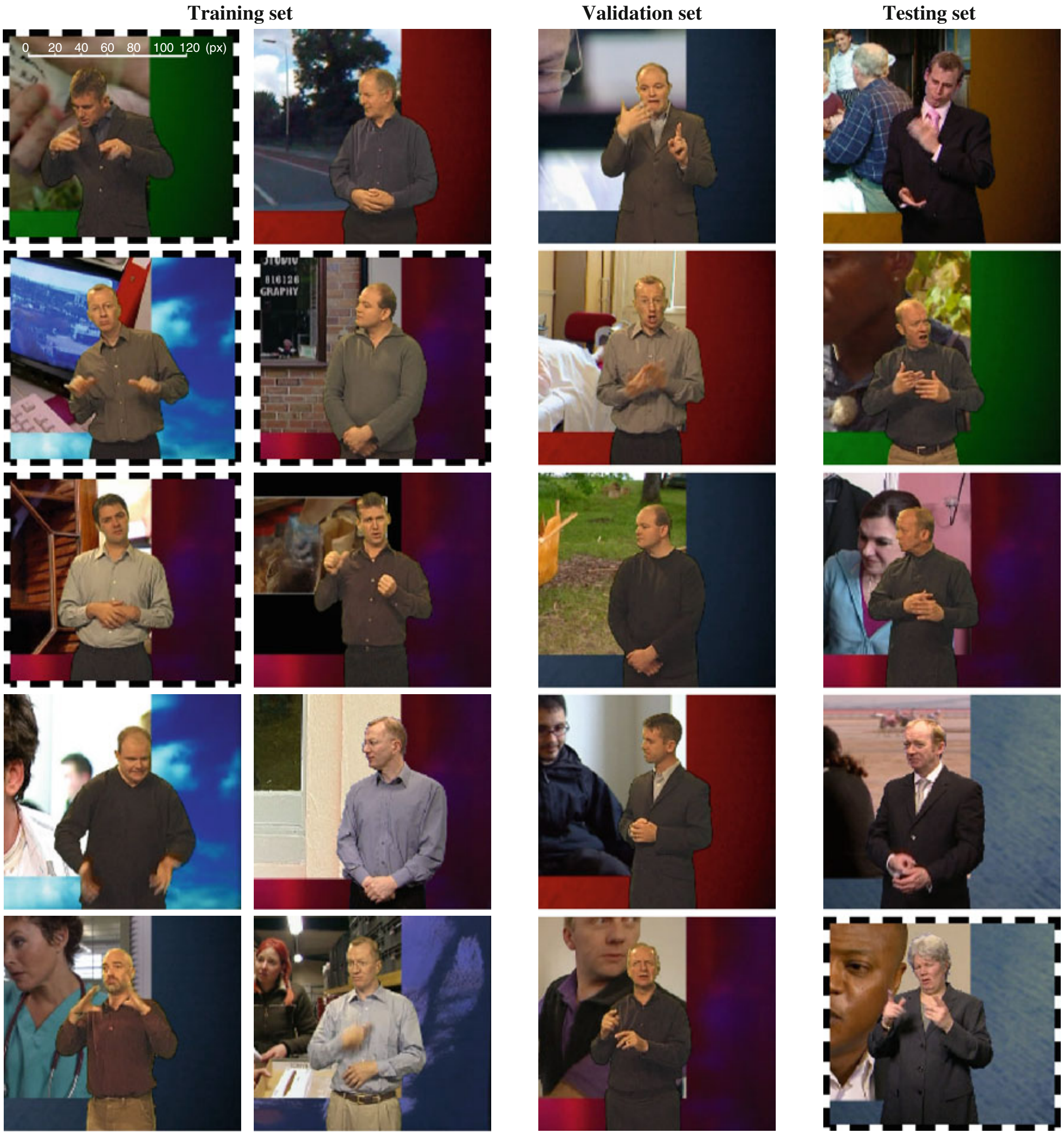
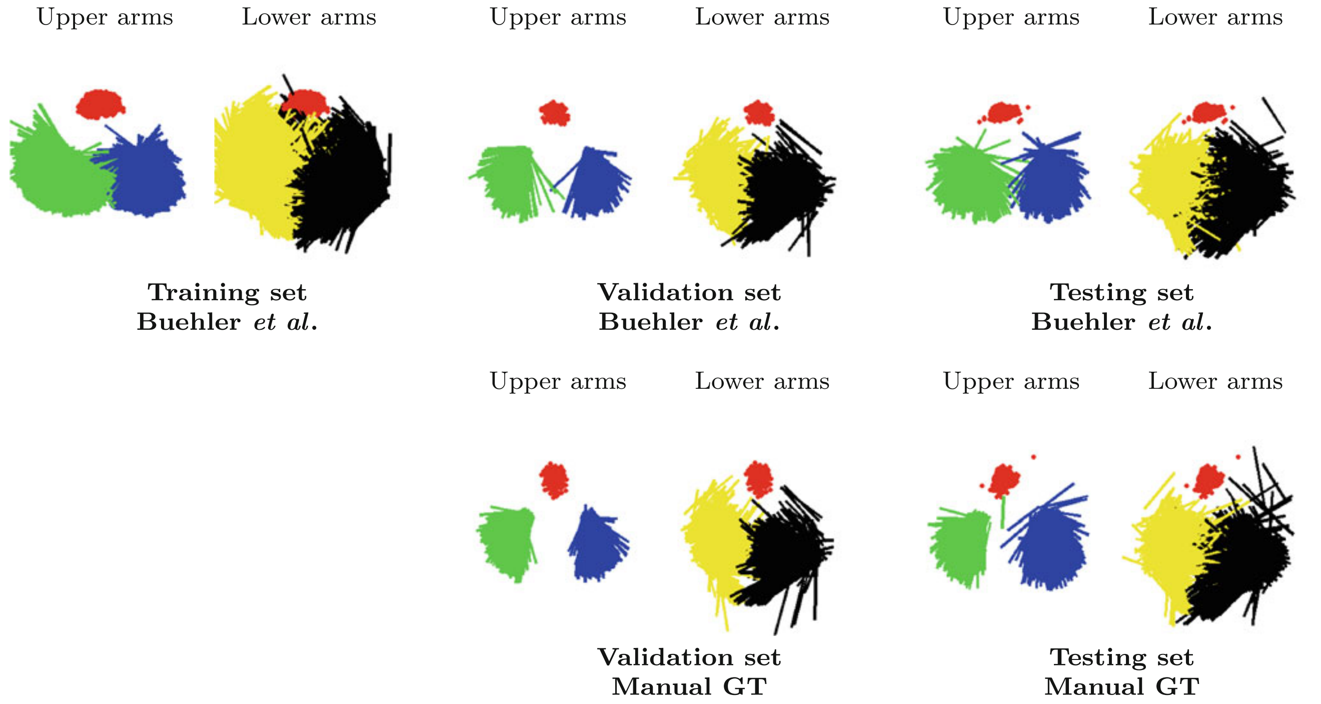


Fig. 11 *Visualisation of complete dataset* showing one example frame per video. Videos are split into training, validation and testing sets. Variation in terms of signer identity, clothing and background video content is ensured in the training set by using different videos and only duplicating signers if they are wearing different clothing. The testing set contains completely different signers than those present in the training or validation sets. Frames with *black dashed border* indicate those videos used for the Random Forest experiments in Sect. 5.3.1. A *scale bar* is provided in the top left hand corner image to compare pixel distance with signer size

5.1.1 Split into Training/Validation/Testing Sets

The full set of 20 videos from our dataset are used. They are split into three disjoint sets: 10 videos for training, 5 for validation and 5 for testing as shown in Fig. 11. Parameters are optimised on the validation set, and the testing set is reserved solely for examining the performance of our system at test time. All videos are recorded using one of 9 different signers. The training and validation set contain five different signers and the testing set another four different signers. Splitting the data this way maintains enough diversity for training but also ensures fairness as the testing set contains completely different signers than the training and validation sets. We maximise the variation in appearance of signers in the training set by only duplicating signers if they are wearing different clothing. Moreover, signers in the validation set all wear different clothing than those in training and testing.



Training set Buehler *et al.*

Validation set Buehler *et al.*

Testing set Buehler *et al.*

Validation set Manual GT

Testing set Manual GT

Fig. 12 *Scatter plots of stickmen* inspired by Tran and Forsyth (2010) show plots of upper and lower arm placements for every frame in the training, validation and testing sets. Poses are normalised to the mid-point between shoulders. Head centre points are rendered as *red dots*, *right* and *left upper arms* are shown as *green* and *blue lines* respectively. *Right* and *left lower arms* are shown as *yellow* and *black lines* respectively. Poses are not scale-normalised, meaning scale and location variation is directly observable between sets. *Top row* illustrates pose outputs from Buehler et al.'s tracker and *bottom row* is from manual ground truth (GT). Manual GT for the training set is not plotted as we do not have labels for all videos (Color figure online)

5.1.2 Pose Sampling and Visualisation

Sampling The random forest and evaluator are trained and tested on frames sampled from each video. Frames are sampled for training by first clustering the training data according to the signers pose (provided by Buehler et al.'s tracker), and uniformly sampling frames across clusters. K-means clustering with 100 clusters is used. Sampling in this way increases the diversity of poses in the training set. This in turn helps the forest generalise to testing data and improves accuracy on unusual poses. For testing and validation videos, 200 frames containing a diverse range of poses are sampled in the same way from each of the 5+5 videos (2,000 frames in total). Sampling the testing data using the same strategy ensures the accuracy of joint estimates are not biased towards poses which occur more frequently, e.g. "resting" poses between signs.

Visualisation A scatter plot of stickmen Tran and Forsyth (2010) is shown in Fig. 12, illustrating upper and lower arm placements for every frame in the training, validation and testing sets. Poses are normalised to the mid-point between shoulders. A wide coverage of different poses obtained from Buehler et al.'s tracker are observed in the training set. Also illustrated are scatter plots for validation and testing sets comparing Buehler et al.'s tracker with manual ground truth. According to Buehler et al.'s tracker, poses in testing frames cover a similar space of poses as in training frames. This demonstrates the effectiveness of our frame sampling method at sampling a diverse range of poses. Comparing scatter plots from manual ground truth with Buehler et al.'s tracker, one can also observe that errors in Buehler et al.'s tracker do make the span of poses look slightly larger.

5.1.3 Ground Truth Labelling

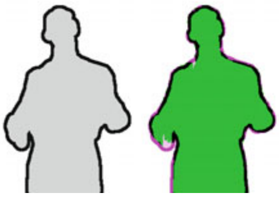
The 200 sampled frames with diverse poses from each of the videos in the validation (5 videos) and testing (another 5 videos) set are manually annotated with joint locations (2,000 frames in total). The validation frames are used for parameter optimisation, and the testing frames are used for evaluating the joint estimates.

5.1.4 Evaluation Measure

In all joint estimation experiments we evaluate the performance of the system by comparing estimated joints against frames with manual ground truth. An estimated joint is deemed correctly located if it is within a set distance of d pixels from a marked joint centre. Accuracy is measured as

Signer	Overlap score	Standard deviation
1	0.962	0.049
2	0.959	0.072
3	0.947	0.043
4	0.960	0.025
5	0.965	0.043
Mean	**0.959**	**0.047**

(a)



(b) **(c)**

Fig. 13 Co-segmentation evaluation using overlap score. **a** Overlap scores for each test signer; **b** example of the ground truth trimap (*white* is background, *grey* is foreground and *black* is unknown); **c** segmentation (*green*) evaluated against the ground truth (*magenta* and *black*) (Color figure online)

the percentage of correctly estimated joints. The experiments use a distance of $d = 5$ pixels from ground truth. A scale superimposed on the top left frame in Fig. 11 shows how pixel distance relates to signer size.

5.2 Co-segmentation

The co-segmentation algorithm is evaluated in two experiments. The first experiment uses ground truth segmentation masks to evaluate the quality of segmentations. The second experiment uses silhouettes rendered based on ground truth joint locations as described in Fig. 7.

5.2.1 Experiment 1: Overlap of Foreground Segmentation with Ground Truth

In this experiment the segmentation masks are compared against manual foreground segmentation ground truth. This ground truth consists of manually labelled foreground segmentation trimaps for 20 frames for each of the five test signers (100 frames in total). The frames are sampled uniformly from different pose clusters (as described in Sect. 5.1). The overlap score from Sect. 4.1 is evaluated separately for each test signer. The mean overlap scores and standard deviations are given in Fig. 13.

5.2.2 Experiment 2: Overlap of Foreground Segmentation with Silhouettes Rendered Based on Joints

In this experiment an overlap score is computed by rendering rectangles at the manual ground truth joint positions as shown in Fig. 7. This is done using the frames in the test and validation sets that have manual ground truth joint locations (Sect. 5.1 above), and is used for evaluating the quality of segmentations for the evaluator. Table 1 shows the attained segmentation overlap scores. A perfect overlap is not expected since the rendered rectangles are only approximations to the true ground truth segmentation. However, as demonstrated in Fig. 8, the overlap score still gives a useful indication of whether the segmentation is good or not. Figure 14 shows the cumulative distribution function of the overlap scores over the test and validation sets. It can be observed that the majority of scores are in the range 0.85–0.95, with no scores below 0.4 or above 0.95, and a small proportion of scores between 0.6 and 0.8. This demonstrates that the segmentation quality score used for the evaluator is fairly accurate.

Table 1 Co-segmentation evaluation using overlap of segmentation and rendered silhouette

Data subset	Avg overlap score	Standard deviation
Test set	0.8628	0.0503
Validation set	0.8542	0.0637

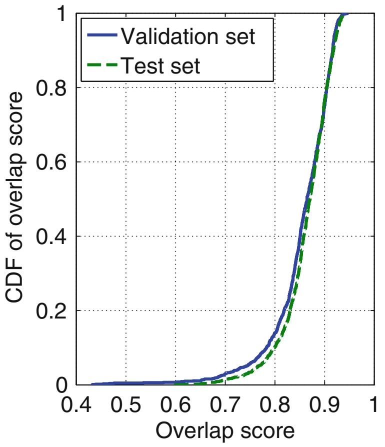


Fig. 14 Cumulative distribution function of segmentation overlap scores

5.3 Random Forest Regression

The joint estimation method is evaluated in four experiments: (i) Frame representation, which explores alternative inputs for the forest and demonstrates the effectiveness of using a segmented colour posterior image (obtained through co-segmentation) over using other simple representations. (ii) Parameter optimisation, which observes the effect of varying the most influential parameters of the random forest. (iii) Increasing training data, where the performance of the random forest is analysed as the amount of training data is increased. (iv) Random forest versus state-of-the-art, where our joint estimation method is pitched against Buehler et al.'s tracker, and pose estimation method of Yang and Ramanan (2011) which uses a mixture of parts.

5.3.1 Experiment 1: Frame Representation

Frames of the videos are represented in one of four different ways: (i) a raw colour pixel representation in LAB (LAB), (ii) colour posterior on the whole image (CP), (iii) signer sil-

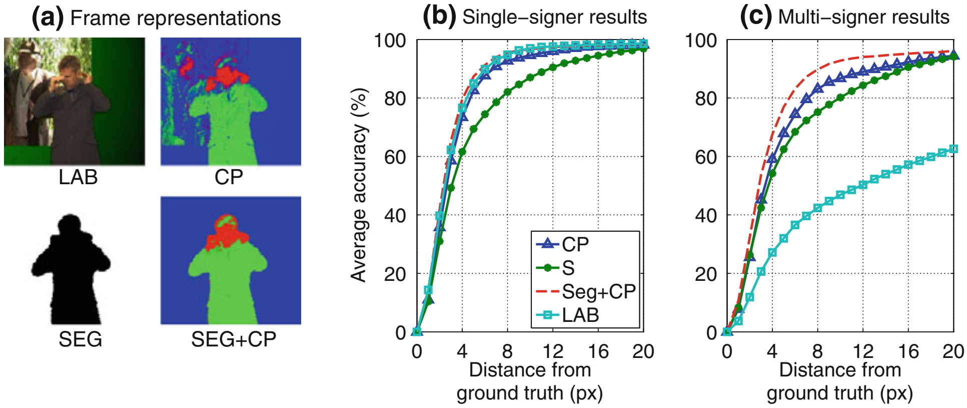


Fig. 15 **a** Example frames showing different methods for representing a frame. **b** Average accuracy of single-signer and **c** multi-signer forests as allowed distance from ground truth is increased. Results for forests trained and tested on different types of frame representation are shown. Using SEG+CP proves best for both single-signer and multi-signer forests

Table 2 Average accuracy of per-joint estimates for single-signer forests measured as 5 pixels from manual ground truth. Using Seg+CP outperforms all other input types

Method	Head	R Wrist	L Wrist	R Elbow	L Elbow	R Shldr	L Shlder	Average
LAB	**98.0**	63.9	**85.8**	67.6	79.2	**87.4**	86.1	81.1
CP	97.7	**70.3**	82.9	67.9	70.0	84.3	72.6	78.0
S	91.9	22.2	30.8	67.8	78.8	82.2	89.0	66.1
Seg + CP	97.6	64.9	84.1	**72.5**	**80.2**	86.8	**92.0**	**82.6**
Buehler et al. (2011)	96.4	58.8	66.0	67.6	71.5	83.1	83.7	75.3

Bold values indicate frame representation with highest accuracy

houette (S), and (iv) segmented colour posterior (Seg+CP), produced through co-segmentation (examples showing each type are shown in Fig. 15a). In this experiment we ascertain the optimal frame representation for producing the most accurate joint estimates. The experiment is conducted in two settings: (1) training and testing on the same signers, as reported by Buehler et al. (2011), and (2) training on multiple signers and testing on an *unseen* signer. This second experiment quantifies the generalisation performance of the forest as the frame representation is altered.

Protocol A sample of five videos from our set of 20 are used in this experiment. Example frames (indicated by a dashed black border) from each of these videos are shown in Fig. 11. We split these videos into two sections: the first 60 % of the video is used for training and the remaining 40 % is used for testing. Five different single-signer forest are trained and tested on each video separately. The data used to train each tree is formed by sampling labelled pixels from the training videos. First 500 diverse frames are sampled and then 500 pixels per frame are chosen (all 91 joint pixels and 409 randomly sampled background pixels). Multi-signer forests are evaluated using fivefold cross validation on videos of 5 different signers, where the RFs are trained on 4 videos and evaluated on a 5th "held-out" video. The data used to train each tree is formed by sampling 1,000 frames across all 4 videos (250 diverse frames per video) and then 500 pixels from each frame.

Results Figure 15 shows average joint estimation accuracy for both single-signer and multi-signer forests as the threshold on allowed distance from manual ground truth is increased. For single-signer forests SEG+CP is on a par with an LAB frame representation, and both perform well. However, for multi-signer forests it can be noticed that using LAB does not generalise well, and performs the worst. On the other hand, SEG+CP maintains best performance in both cases.

CP loses accuracy when going from the single-signer case to a multi-signer case. The failures are due to changes in background video content neighbouring the right joints of the signer. Tables 2 and 3 show the average accuracy per joint for single-signer and multi-signer forests respectively, using an allowed distance from ground truth of $d = 5$ pixels. In the case of CP there is only a small drop in left-wrist accuracy between the multi-signer and single-signer forests. This is due to the left wrist being shielded from the dynamic background by the signers largely unchanging body appearance.

Removing the background content and using SEG+CP allows the forest to learn a more refined appearance of body joints and boost detection accuracy by reducing the influence of noise. However, the method is left at the mercy of the background removal procedure. One such failure case for SEG+CP occurs when the segmentation cuts off a hand confusing it for background content, causing CP to outperform SEG+CP for the right-wrist in the single-signer case.

Table 3 Average accuracy of per-joint estimates for multi-signer forests trained and tested on a subset of the dataset as described in Sect. 5.3.2. Estimates are deemed correct if they are within 5 pixels of manual ground truth

Method	Head	R Wrist	L Wrist	R Elbow	L Elbow	R Shldr	L Shlder	Average
LAB	56.8	7.6	14.8	22.8	37.4	36.8	47.8	32.0
CP	93.8	52.9	**80.4**	30.8	62.1	75.7	79.4	67.9
S	88.4	15.6	18.4	59.8	**78.6**	85.0	91.4	62.5
Seg + CP	95.0	**60.3**	80.0	57.3	63.4	**88.0**	**94.5**	**76.9**
Buehler et al. (2011)	**96.4**	58.8	66.0	**67.6**	71.5	83.1	83.7	75.3

Bold values indicate frame representation with highest accuracy

The next experiment explores parameter tuning for the SEG+CP frame representation. We discover the effect forest parameters have on accuracy when using a large training set with more variation in both appearance and signers poses.

5.3.2 Experiment 2: Parameter Tuning

This experiment fully analyses the effect tree depth, number of trees in the forest and size of the sliding window have on joint estimation accuracy, and the sorts of parameter settings one should expect to use for optimal performance. Only one parameter is analysed at a time with the remaining fixed. Fixed values used are a tree depth of 32, sliding window width of 71 pixels and a forest of 8 trees.

Protocol Multi-signer forests are trained using all 10 training videos. Training data for each tree is formed by sampling as described in Sect. 5.1 from each video and sampling 700 pixels per frame (91 joint pixels + 609 background pixels) amounting to 3.5 million data points per tree. Forests are retrained for each parameter setting and tested on 1,000 ground truth frames in the validation set.

Results: Tree Depth Figure 17a–c shows the effect tree depth, number of trees in forest and sliding window width have on the joint estimation accuracy respectively. Accuracy per joint, averaged over left and right body parts, is plotted. In Fig. 17a a steady increase in accuracy is observed as tree depth increases from 4 to 32. Beyond depth 32 the accuracy starts dropping. This drop in accuracy is due to overfitting and occurs for all but the wrist joints as depth is increased further. For wrists an optimal depth at 64 is found, implying the wrists' appearance and context are much more varied than other body joints, with classification requiring many more tests. This result also suggests that a single class forest per joint, optimised with different parameter settings, may produce better overall accuracy.

Figure 16 visualises the output joint distributions (see joint colour key in Fig. 16a) as tree depth is increased. With low depth the forest generally splits large portions of the image into probable joint labels. Joint confidences are weak and 'wash' together. As tree depth increases, confidences become higher for a particular spatial location.

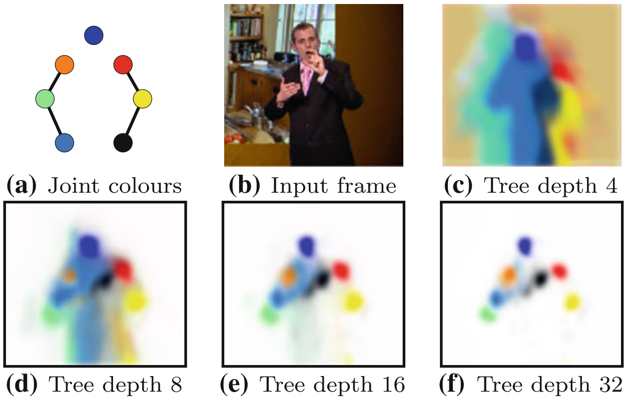


Fig. 16 Visualisation of forest output when applied to example input frame (**b**). The output confidence map per joint label as tree depth increases from 4 to 32 is shown in **c**–**f**. Higher intensity colour implies higher probably of a joint label—key shown in (**a**) (Color figure online)

Results: Number of Trees For all joints, adding more trees to the forest produces higher accuracy. Up to 8 trees were tested and the plot in Fig. 17b indicates more trees could further improve performance.

Results: Window Width The forest draws tests from within a sliding window centred on the pixel that is being classified. By adjusting the size of the sliding window one can control the amount of context used for classification. Context is important because it provides information about the relative placement of body joints, such as that shoulders are found below the head. The plot in Fig. 17c reveals an increase in accuracy as window width size is increased from 31 to 71 pixels. A decline in accuracy is observed as the window width is increased further. There are two possible reasons for this behaviour: (1) There are not enough test samples being drawn as the window width is increased past 71 pixels. (2) Overfitting occurs and a prior on the types of pose seen during training is being learnt.

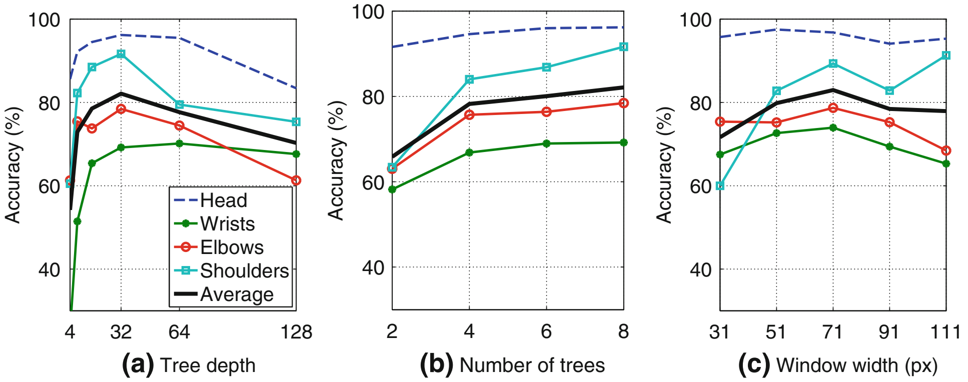


Fig. 17 Accuracy of random forest as **a** tree depth, **b** number of trees in forest and **c** sliding window width are adjusted

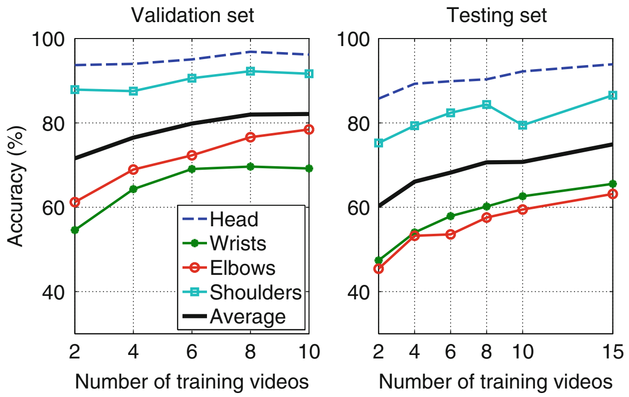


Fig. 18 Forest performance as amount of training data is increased. Results on validation set and testing set are shown

5.3.3 Experiment 3: Increasing Training Data

This experiment tests the intuition that more training data will improve generalisation of the forest and hence increase the accuracy of joint estimates.

Protocol Multiple forests are trained, each using a sample of 2 videos from the set of 10 training videos. The SEG+CP frame representation and multi-signer forests are used. Forest parameters are optimised by maximising the average forest accuracy when applied to the 1,000 ground truth frames in the validation set. This process is repeated for a sample of 4, 6, 8 and 10 training videos. The number of forests trained for each sample size is proportional to the total number of possible sample combinations (where the proportion constant is $\frac{1}{21}$). E.g. for a sample size of 4 videos, we average over $\left\lceil \left({}^{10}C_4\right)/21 \right\rceil = 10$ videos. For a sample size of 2, 4, 6, 8 and 10 videos, we averaged over 3, 10, 10, 3 and 1 forest(s) respectively. Finally we also train a forest with 15 videos using the testing and validation sets combined. For this forest, we are not able to tune parameters due to a limited number of available videos. We therefore fix them at the optimal parameters found when training with 10 videos. Seven hundred pixels per frame are sampled from 500 diverse frames extracted from each of the sampled videos. All forests are tested on 1,000 ground truth frames from videos in the testing set.

Results Figure 18a shows the average accuracy achieved by forests on the validation set. For all joint types we observe a general increase in accuracy as more training data is added. The same trend is observed when applying these forests to unseen signers in the testing set as shown in Fig. 18b. Of particular interest is the drop in accuracy of the shoulder joints when going from 8 to 10 videos. We believe this is due to a particular video having noisy segmentations on the signer's left shoulder. It can also be noticed that elbows have higher accuracy than wrists in the validation set, but *vice versa* on the testing set. This is due to more segmentation errors at elbow locations in the testing videos.

5.3.4 Random Forest Versus State-of-the-Art

In this experiment the random forest is compared to Buehler et al.'s tracker and the deformable part based model by Yang and Ramanan (2011).

Protocol The forest is trained on the full 15 video training set. The optimal parameters from Sect. 5.3.2 are used, i.e. a tree depth of 32, window size of 71 and 8 trees. The model by Yang and Ramanan (2011) is trained for two different types of video input: (1) The original RGB input, and (2) an RGB input with the background content removed by setting it to black. For both types of input the full 15 video dataset is used for training. From each video 100 diverse training frames were sampled, totaling 1,500 frames. Model parameters were set the same as those used for upper body pose estimation in Yang and Ramanan (2011). Negative training images not containing people were taken from the INRIA dataset. Testing for all three upper body pose estimators is conducted on the full 5 video testing set.

Results Figure 19 shows accuracy as the allowed distance from ground truth is increased. The head accuracy and aver-

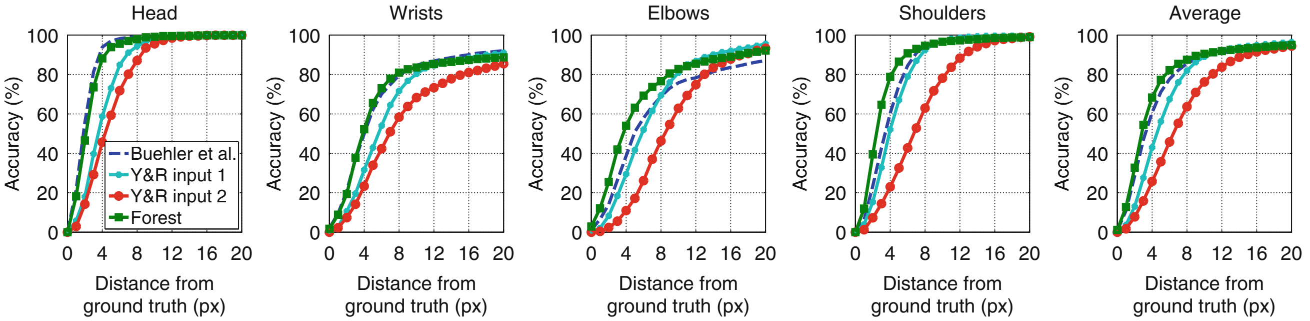


Fig. 19 Comparison of joint tracking accuracy of random forest trained on 15 videos against Buehler et al.'s tracker and Yang and Ramanan's pose estimation algorithm. Plots show accuracy per joint type (averaged over left and right body parts) as allowed distance from manual ground truth is increased

Table 4 Average accuracy of per joint estimates on the full 5 video testing set. Estimates are deemed correct if they are within 5 pixels of manual ground truth

Method	Head	R Wrist	L Wrist	R Elbow	L Elbow	R shldr	L Shlder	Average
Yang and Ramanan (2011) input 1	73.1	39.4	46.4	38.8	44.5	57.8	76.2	53.7
Yang and Ramanan (2011) input 2	59.3	28.3	39.6	15.2	19.1	46.4	18.7	32.4
Buehler et al. (2011)	**97.0**	53.9	70.6	41.6	60.2	73.8	75.1	67.5
Random forest	93.9	**59.5**	**71.6**	**58.8**	**67.5**	**80.1**	**93.0**	**74.9**

Bold values indicate method with highest accuracy

age accuracy over left and right joints are plotted. For all joints but the head, the forest consistently performs better than Buehler et al.'s tracker. For the wrists and shoulders, erroneous joint predictions by the forest are further from the ground truth than erroneous predictions from Buehler et al.'s tracker once joint predictions are at least ≈ 10 pixels from ground truth. This fact means that it is likely to be easier for a pose evaluator to detect errors made by the forest. Interestingly, the model by Yang and Ramanan (2011) achieved best performance when using the original RGB video input (input 1) over using a background removed version (input 2). We suggest that this is due to a poor representation of negative image patches in input 2 when using negative training images from the INRIA dataset. Overall, Yang and Ramanan's model is the least accurate over all joint types.

Table 4 shows per joint accuracy for Buehler et al.'s tracker and the forest using an allowed distance from ground truth of $d = 5$ pixels. The forest performs best with an average accuracy of 74.9 %. This suggests noisy data from Buehler et al.'s tracker is smoothed over by more consistent data at the leaf nodes of the trees. Results for the forest on an example 5 frames from the testing set is shown qualitatively in Fig. 20.

5.4 Pose Evaluator

The pose evaluator is assessed here on the ability to label joint predictions per frame as either success or fail. The quality of joint predictions on success frames is also used as a measure of the evaluator's performance.

Protocol The evaluator is trained on the validation set and tested on the test set shown in Fig. 11. For training, the joint tracking output from Buehler et al. (2011) is used to automatically label poses for a set of training frames as success or fail. For testing, the 1,000 frames with manual ground truth (described in Sect. 5.3.2) are used.

Results: Choice of Operating Point Figure 21a shows the ROC curve of the evaluator when varying the operating point (effectively changing the threshold of the SVM classifier's decision function). This operating point determines the sensitivity at which the evaluator discards frames. The optimal operating point occurs at a point on the curve which best trades off false positives against true positives. This is a point closest to the top left hand corner of the plot. To gain further insight into the effect of the operating point choice on joint estimates, we plot this value against joint prediction accuracy in Fig. 21b. This illustrates the correlation between the SVM score and percentage of frames that the evaluator marks as successes (i.e. not failures). One can observe that when keeping the top 10 % frames, a 90 % average accuracy could be attained. More frames can be kept at the cost of loss in average accuracy. The bump at 0.8 suggests that at a particular SVM score, the pose evaluator begins to remove some frames which may not contain a higher degree of error compared to frames removed with a higher SVM score threshold. How-

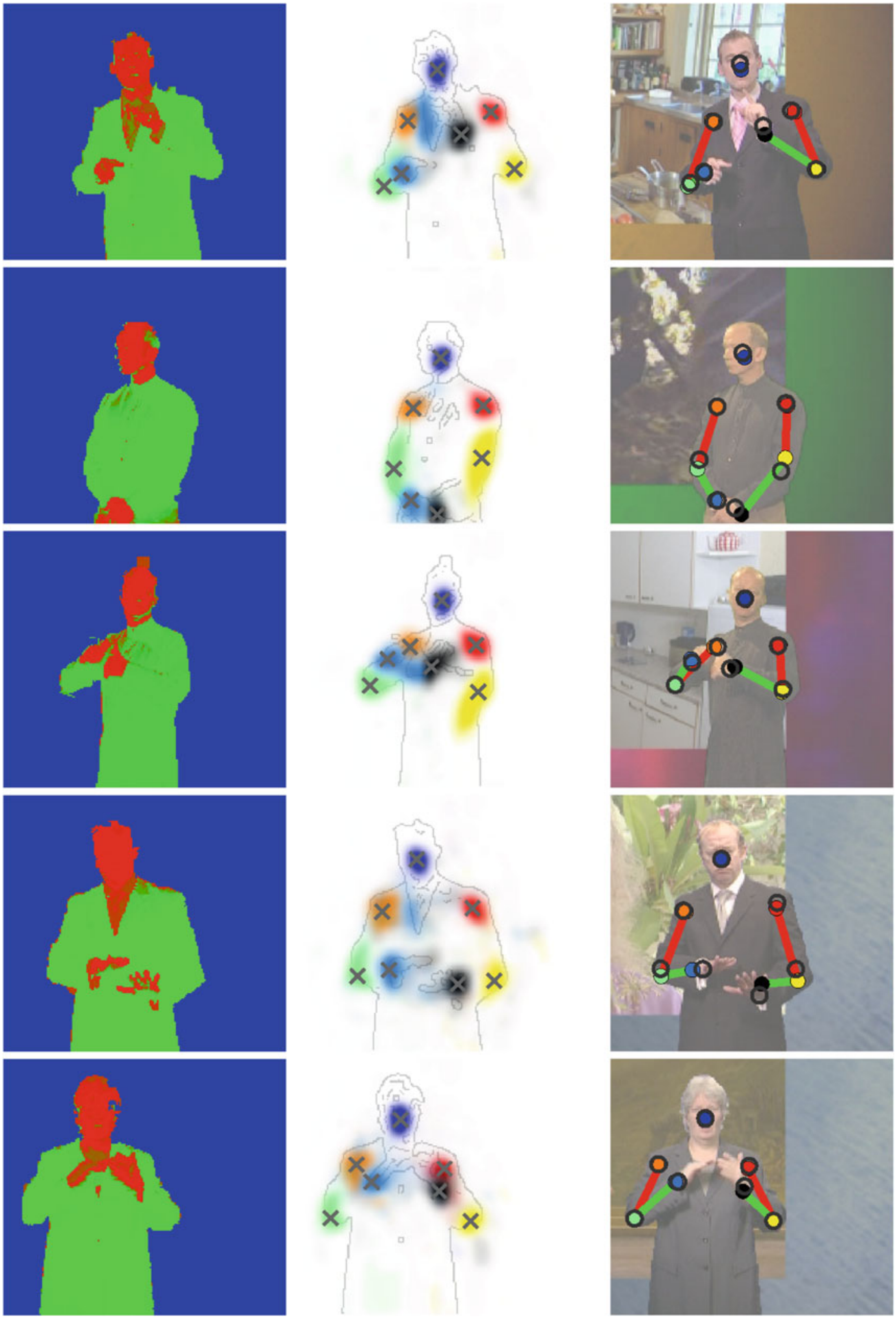


Fig. 20 Joint estimation results. *Left* shows colour model images, from which we obtain probability densities of joint locations shown on top of the colour model edge image in *centre*. Different colours are used per joint (higher intensity colour implies higher probability). Maximum probability per joint is shown as grey crosses. *Right* shows a comparison of estimated joints (*filled in circles* linked by a skeleton are) overlaid on faded original frame, with ground truth joint locations (*open circles*)

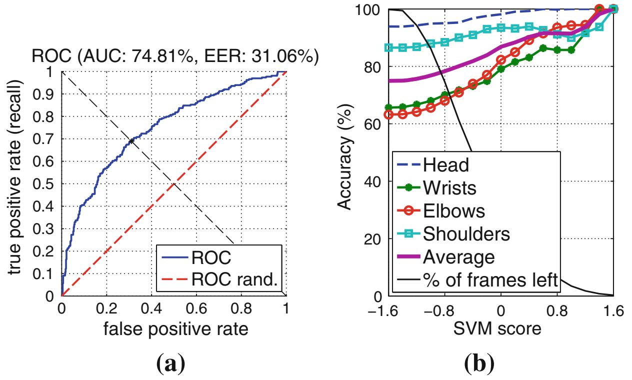


Fig. 21 Pose evaluator classification performance. **a** ROC curve of the evaluator. **b** Change in accuracy as a function of the percentage of frames left after discarding frames that the evaluator detects as failures. For **b** the accuracy threshold is set as 5 pixels from manual ground truth

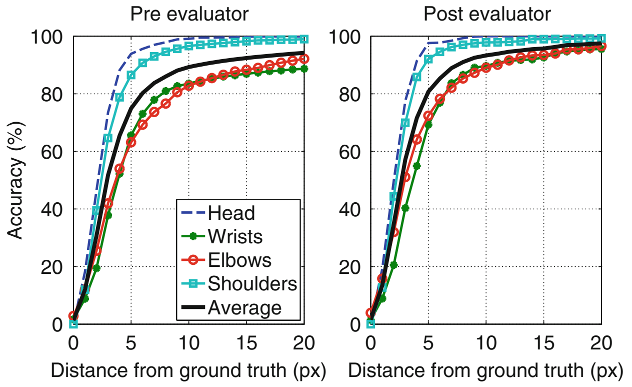


Fig. 22 Average accuracy of per-joint estimates without (*left*) and with (*right*) evaluator when the operating point of the pose evaluator is set to the optimum in Fig. 21a

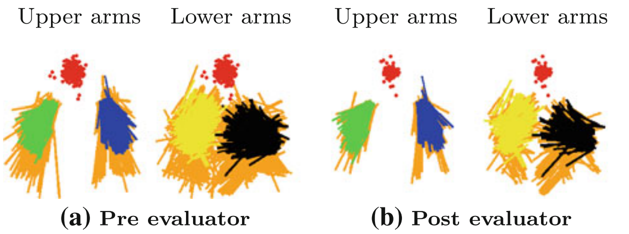


Fig. 23 **a** Shows scatter plots of stickmen for pose estimates from forest on all training data. **b** Shows scatter plot of pose estimates from forest on training data marked as containing good poses by the evaluator. Elbow and wrist joints greater than 5px from ground truth are indicated by *orange sticks* (Color figure online)

ever, in general there is a positive correlation between the SVM score and pose estimation accuracy.

Results: Joint Localisation Figure 22 demonstrates the improvement in joint localisation obtained by discarding frames that the evaluator classifies as failed. This yields an 8.5 % increase in average accuracy (from 74.9 to 83.4 %) at a maximum distance of 5 pixels from ground truth, with 40.4 % of the test frames remaining. One can observe a particularly significant improvement in wrist and elbow localisation accuracy. This is due to a majority of hand mixup frames being correctly identified and filtered away. The improvements in other joints are due to the evaluator filtering away many frames where joints are assigned incorrectly due to segmentation errors.

Results: Pose Visualisation A scatter plot of stickmen for the forest joint predictions are plotted on all test frames in Fig. 23a. Sticks are marked as orange if the elbow or wrist joints are more than 5 pixels from ground truth. One observes erroneous joint predictions tend to exaggerate the length of upper arms. Typically wrist joint errors occur when the wrists are further away from the torso centre. Figure 23b shows the same plot as in Fig. 23a but only on testing frames marked as successful by the evaluator. Notice the evaluator has successfully removed errors on the elbows and wrists while still retaining the majority of the correct poses.

5.5 Computation Time

The following computation times are on a 2.4 GHz Intel Quad Core I7 CPU with a 320 × 202 pixel image. The computation time for one frame is 0.14 s for the co-segmentation algorithm, 0.1 s for the random forest regressor and 0.1 s for the evaluator, totalling 0.21 s (≈ 5fps). Face detection Zhu and Ramanan (2012) takes about 0.3 s/frame for a quad-core processor. The per-frame initialisation timings of the co-segmentation algorithm are 6 ms for finding the dynamic background layer and static background, 3 ms for obtaining a clean plate and 5 ms for finding the image sequence-wide foreground colour model, totalling 14 ms (approx. 24 min for a 100 K frames). In comparison, Buehler et al.'s method runs at 100 s per frame on a 1.83 GHz CPU, which is two orders of magnitude slower. Each tree for our multi-signer RFs trained with 15 videos takes 20 h to train.

6 Conclusion

We have presented a fully automatic arm and hand tracker that detects joint positions over continuous sign language video sequences of more than an hour in length. Our framework attains superior performance to a state-of-the-art long term tracker Buehler et al. (2011), but does not require the manual annotation and, after automatic initialisation, performs tracking in real-time on people that have not been seen during training. Moreover, our framework augments the joint estimates with a failure prediction score, enabling incorrect poses to be filtered away. Future work includes improving the evaluator by adding new features, and using its output not only as an indication of failure but also as an evaluation measure to help correct failed poses.

Acknowledgments We are grateful to Lubor Ladicky for discussions, and to Patrick Buehler for his very generous help. Funding is provided by the Engineering and Physical Sciences Research Council (EPSRC) grant Learning to Recognise Dynamic Visual Content from Broadcast Footage.

References

Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, *9*(7), 1545–1588.

Andriluka, M., Roth, S., & Schiele, B. (2012). Discriminative appearance models for pictorial structures. *International Journal of Computer Vision*, *99*(3), 259–280.

Apostoloff, N. E., & Zisserman, A. (2007). Who are you?—real-time person identification. In *Proceedings of the British machine vision conference*.

Benfold, B., & Reid, I. (2008). Colour invariant head pose classification in low resolution video. In *Proceedings of the British machine vision conference*.

Bosch, A., Zisserman, A., & Munoz, X. (2007). Image classification using random forests and ferns. In *Proceedings of the international conference on computer vision*.

Bowden, R., Windridge, D., Kadir, T., Zisserman, A., & Brady, J. M. (2004). A linguistic feature vector for the visual interpretation of sign language. In *Proceedings of the European conference on computer vision*. Berlin: Springer.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Buehler, P., Everingham, M., Huttenlocher, D. P., & Zisserman, A. (2011). Upper body detection and tracking in extended signing sequences. *International Journal of Computer Vision*, *95*(2), 180–197.

Buehler, P., Everingham, M., & Zisserman, A. (2009). Learning sign language by watching TV (using weakly aligned subtitles). In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Buehler, P., Everingham, M., & Zisserman, A. (2010). Employing signed TV broadcasts for automated learning of British sign language. In *Workshop on representation and processing of sign languages*.

Chai, Y., Lempitsky, V., & Zisserman, A. (2011). BiCoS: A bi-level co-segmentation method for image classification. In *Proceedings of the international conference on computer vision*.

Chai, Y., Rahtu, E., Lempitsky, V., Van Gool, L., & Zisserman, A. (2012). Tricos: A tri-level class-discriminative co-segmentation method for image classification. In *European conference on computer vision*.

Charles, J., Pfister, T., Magee, D., Hogg, D., & Zisserman, A. (2013). Domain adaptation for upper body pose tracking in signed TV broadcasts. In *Proceedings of the British machine vision conference*.

Chunli, W., Wen, G., & Jiyong, M. (2002). A real-time large vocabulary recognition system for Chinese Sign Language. *Gesture and sign language in HCI*.

Cooper, H., & Bowden, R. (2007). Large lexicon detection of sign language. *Workshop on human computer interaction*.

Cooper, H., & Bowden, R. (2009). Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Cootes, T., Ionita, M., Lindner, C., & Sauer, P. (2012). Robust and accurate shape model fitting using random forest regression voting. In *Proceedings of the European conference on computer vision*.

Criminisi, A., Shotton, J., & Konukoglu, E. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, *7*(2), 81–227.

Criminisi, A., Shotton, J., & Robertson, & D., Konukoglu, E., (2011). Regression forests for efficient anatomy detection and localization in CT studies. In *International conference on medical image computing and computer assisted intervention workshop on probabilistic models for medical image analysis*.

Dalal, N., & Triggs, B. (2005). Histogram of Oriented Gradients for Human Detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Dantone, M., Gall, J., Fanelli, G., & Van Gool, L. (2012). Real-time facial feature detection using conditional regression forests. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Dreuw, P., Deselaers, T., Rybach, D., Keysers, D., & Ney, H. (2006). Tracking using dynamic programming for appearance-based sign language recognition. In *Proceedings of the IEEE conference on automatic face and gesture recognition*.

Dreuw, P., Forster, J., & Ney, H. (2012). Tracking benchmark databases for video-based sign language recognition. In *Trends and topics in computer vision* (pp. 286–297). Berlin: Springer.

Eichner, M., & Ferrari, V. (2009). Better appearance models for pictorial structures. In *Proceedings of the British machine vision conference*.

Eichner, M., Marin-Jimenez, M., Zisserman, A., & Ferrari, V. (2012). 2D articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*, 1–25.

Fanelli, G., Dantone, M., Gall, J., Fossati, A., & Van Gool, L. (2012). Random forests for real time 3D face analysis. *International Journal of Computer Vision*, *101*(3), 1–22.

Fanelli, G., Gall, J., & Van Gool, L. (2011). Real time head pose estimation with random regression forests. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Farhadi, A., & Forsyth, D. (2006). Aligning asl for statistical translation using a discriminative word model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Farhadi, A., Forsyth, D., & White, R. (2007). Transfer learning in sign language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Felzenszwalb, P., Girshick, R., & McAllester, D. (2010). Cascade object detection with deformable part models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Felzenszwalb, P., & Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*(1), 55–79.

Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Ferrari, V., Marin-Jimenez, M., & Zisserman, A. (2008). Progressive search space reduction for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Gall, J., & Lempitsky, V. (2009). Class-specific hough forests for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Geremia, E., Clatz, O., Menze, B., Konukoglu, E., Criminisi, A., & Ayache, N. (2011). Spatial decision forests for MS lesion segmentation in multi-channel magnetic resonance images. *NeuroImage*, *57*(2), 378–390.

Girshick, R., Shotton, J., Kohli, P., Criminisi, A., & Fitzgibbon, A. (2011). Efficient regression of general-activity human poses from depth images. In *Proceedings of the international conference on computer vision*.

Hochbaum, D., & Singh, V. (2009). An efficient algorithm for co-segmentation. In *Proceedings of the international conference on computer vision*.

Jammalamadaka, N., Zisserman, A., Eichner, M., Ferrari, V., & Jawahar, C. V. (2012). Has my algorithm succeeded? An evaluator for human pose estimators. In *Proceedings of the European conference on computer vision*.

Johnson, S., & Everingham, M. (2009). Combining discriminative appearance and segmentation cues for articulated human pose estimation. In *IEEE international workshop on machine learning for vision-based motion analysis*.

Jojic, N., & Frey, B. (2001). Learning flexible sprites in video layers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Joulin, A., Bach, F., & Ponce, J. (2010). Discriminative clustering for image co-segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Kadir, T., Bowden, R., Ong, E., & Zisserman, A. (2004). Minimal training, large lexicon, unconstrained sign language recognition. In *Proceedings of the British machine vision conference*.

Kadir, T., Zisserman, A., & Brady, J. M. (2004). An affine invariant salient region detector. In *Proceedings of the European conference on computer vision*.

Kontschieder, P., Bulò, S., Criminisi, A., Kohli, P., Pelillo, M., & Bischof, H. (2012). Context-sensitive decision forests for object detection. In *Advances in neural information processing systems*.

Kumar, M. P., Torr, P. H. S., & Zisserman, A. (2008). Learning layered motion segmentations of video. *International Journal of Computer Vision*, *76*, 301–319.

Lepetit, V., & Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(9), 1465–1479.

Liu, C., Gong, S., Loy, C., & Lin, X. (2012). Person re-identification: What features are important?. In *Proceedings of the European conference on computer vision*.

Marée, R., Geurts, P., Piater, J., & Wehenkel, L. (2005). Random subwindows for robust image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Moeslund, T. (2011). *Visual analysis of humans: Looking at people*. Berlin: Springer.

Nowozin, S., Rother, C., Bagon, S., Sharp, T., Yao, B., & Kohli, P. (2011). Decision tree fields. In *Proceedings of the international conference on computer vision*.

Ong, E., & Bowden, R. (2004). A boosted classifier tree for hand shape detection. In *Proceedings of the international conference on automatic face and gesture recognition*.

Ozuysal, M., Calonder, M., Lepetit, V., & Fua, P. (2010). Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(3), 448–461.

Pfister, T., Charles, J., Everingham, M., & Zisserman, A. (2012). Automatic and efficient long term arm and hand tracking for continuous sign language TV broadcasts. In *Proceedings of the British machine vision conference*.

Pfister, T., Charles, J., & Zisserman, A. (2013). Large-scale learning of sign language by watching TV (using co-occurrences). In *Proceedings of the British machine vision conference*.

Ramanan, D. (2006). Learning to parse images of articulated bodies. In *Advances in neural information processing systems*.

Ramanan, D., Forsyth, D. A., & Zisserman, A. (2007). Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(1), 65–81.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: interactive foreground extraction using iterated graph cuts. In *Proceedings of the ACM SIGGRAPH conference on computer graphics*.

Rother, C., Minka, T., Blake, A., & Kolmogorov, V. (2006). Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Santner, J., Leistner, C., Saffari, A., Pock, T., & Bischof, H. (2010). Prost: Parallel robust online simple tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Sapp, B., Jordan, C., & Taskar, B. (2010). Adaptive pose priors for pictorial structures. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Sapp, B., Weiss, D., & Taskar, B. (2011). Parsing human motion with stretchable models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Sharp, T. (2008). Implementing decision trees and forests on a GPU. In *Proceedings of the European conference on computer vision*.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., et al. (2011). Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Shotton, J., Johnson, M., & Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Sivic, J., Zitnick, C. L., & Szeliski, R. (2006). Finding people in repeated shots of the same scene. In *Proceedings of the British machine vision conference*, Edinburgh.

Starner, T., Weaver, J., & Pentland, A. (1998a). Real-time american sign language recognition using desk- and wearable computer-based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), 1371–1375.

Starner, T., Weaver, J., & Pentland, A. (1998b). Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), 1371–1375.

Sun, M., Kohli, P., & Shotton, J. (2012). Conditional regression forests for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Szeliski, R., Avidan, S., & Anandan, P. (2000). Layer extraction from multiple images containing reflections and transparency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Taylor, J., Shotton, J., Sharp, T., & Fitzgibbon, A. (2012). The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tran, D., & Forsyth, D. (2010). Improved human parsing with a full relational model. In *Proceedings of the European conference on computer vision*.

Vogler, C., & Metaxas, D. (1998). ASL recognition based on a coupling between HMMs and 3D motion analysis. In *Proceedings of the international conference on computer vision*.

Yang, Y., & Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yin, P., Criminisi, A., Winn, J., & Essa, I. (2007). Tree-based classifiers for bilayer video Segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Zhu, X., & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Zisserman, A., Winn, J., Fitzgibbon, A., van Gool, L., Sivic, J., Williams, C., & Hogg, D. (2012). In memoriam: Mark Everingham. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2081–2082.