THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

# 2D Articulated Human Pose Estimation and Retrieval in (Almost) Unconstrained Still Images

**Citation for published version:**
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*, vol. 99, no. 2, pp. 190-214. https://doi.org/10.1007/s11263-012-0524-9

**Digital Object Identifier (DOI):**
10.1007/s11263-012-0524-9

**Link:**
Link to publication record in Edinburgh Research Explorer

**Document Version:**
Peer reviewed version

**Published In:**
International Journal of Computer Vision

**General rights**
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 21. Mar. 2026

# 2D Articulated Human Pose Estimation and Retrieval in (Almost) Unconstrained Still Images

**M. Eichner, M. Marin-Jimenez, A. Zisserman, V. Ferrari**

**Abstract** We present a technique for estimating the spatial layout of humans in still images – the position of the head, torso and arms. The theme we explore is that once a person is localized using an upper body detector, the search for their body parts can be considerably simplified using weak constraints on position and appearance arising from that detection. Our approach is capable of estimating upper body pose in highly challenging uncontrolled images, without prior knowledge of background, clothing, lighting, or the location and scale of the person in the image. People are only required to be upright and seen from the front or the back (not side).

We evaluate the stages of our approach experimentally using ground truth layout annotation on a variety of challenging material, such as images from the PASCAL VOC 2008 challenge and video frames from TV shows and feature films.

We also propose and evaluate techniques for searching a video dataset for people in a specific pose. To this end, we develop three new pose descriptors and compare their classification and retrieval performance to two baselines built on state-of-the-art object detection models.

**Keywords** articulated human pose estimation search retrieval

M. Eichner
ETH Zurich
E-mail: eichner(AT)vision.ee.ethz.ch

M. Marin-Jimenez
University of Cordoba
E-mail: mjmarin(AT)uco.es

A. Zisserman
University of Oxford
E-mail: az(AT)robots.ox.ac.uk

V. Ferrari
University of Edinbourgh
E-mail: vferrari(AT)staffmail.ed.ac.uk

## 1 Introduction

Our goal is to automatically detect and estimate the 2D pose of humans in images under the uncontrolled imaging conditions typical of amateur photos, movies and TV shows (fig. 1). Such images are often very cluttered and people may appear at any scale; illumination varies over a diverse palette of lighting conditions; contrast may be poor and motion blur from camera movement (in videos) or shake (in photos) can also be present (fig. 8, 12, 19, 21). A person's appearance is unconstrained, as she can wear any kind of clothing, short or long sleeves, and any colors/textures. In this work 2D pose refers to the image position of the head, torso and limbs.

There are numerous reasons why detecting humans and obtaining their pose is useful. A fundamental one is that often the pose, or a pose sequence, characterizes a person's attitude or action. More generally, applications range from video understanding and search through to surveillance. Indeed 2D human segmentation is often the first step in determining 3D human pose from individual frames [1, 26].

In this work we focus on human upper-bodies only, as they convey the majority of information necessary to recognize the actions carried out by a person. Moreover, TV shows and feature films usually consist of close-ups or medium-shots where legs stay outside the visible frame.

The method we present is for general still images as it requires no prior knowledge of the background, clothing, lighting, or the location and scale of the people in the image. We assume very little about the pose of a person – only that they are upright and in near frontal or rear viewpoints. This requirement is rather weak, as the vast majority of people in photographs and movies appear upright. Importantly, there is no constraint on the pose of the arms. The main theme of the paper is that from an upper body detection we can derive valuable information about the layout and appearance of the person, and this can be employed to reduce the search for their 2D pose (and consequently increase the chances of a

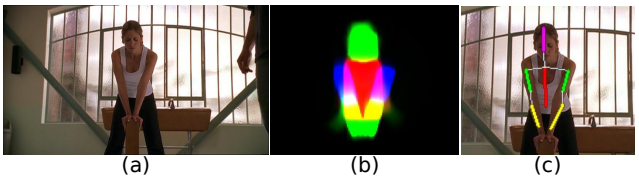



**Fig. 1 Objective of this work.** (a) Input image. (b) Soft-labeling of pixels to body parts or background. Red indicates torso, blue upper arms, green lower arms and head. Brighter pixels are more likely to belong to a part. Color planes are added up, so that yellow indicates overlap between lower-arm and torso, purple between upper-arm and torso, and so on. (c) Stickman representation of pose, obtained by fitting straight line segments to the segmentations in (b). For enhanced visibility, the lower arms are in yellow and the head is in purple.

correct estimate). For example, we know the scale and approximate location of the head and torso. Moreover, we can estimate good person- and image-specific appearance models, which is very important for pictorial structures whose success depends critically on having good appearance models.

The proposed method supports a variety of poses, such as arms folded over the torso or stretching outwards. Starting from the estimated pose, various pose representations can be derived, such as a soft-labeling of every pixel as belonging to a particular body part or the background (fig. 1b); or the 'stickman' [10] of figure 1c, indicating the location, orientation, and size of body parts.

As an application of human pose estimation (HPE), we present here a retrieval system based on poses, which is able to retrieve video shots containing a particular pose from a data set of videos. The pose is specified by a single query frame and we can retrieve shots containing that pose for different people, lighting, clothing, scale and backgrounds. Being able to search video material by pose provides yet another access mechanism over searching for shots containing a particular object or location [60], person [4, 43, 61], action [7, 40], object category or scene category (e.g. indoors/outdoors).

The paper has two main parts. In sections 2 to 8 we present and evaluate our human pose estimation algorithm, and then in the second part we focus on an application called *pose search* (sec. 9 onwards). In section 2 we summarize previous works related to HPE and define concepts on which we build. Section 3 describes our model and outlines the processing pipeline. The building blocks of our HPE approach are detailed in sections 4 to 7. In section 8 we undertake a comprehensive evaluation of the HPE algorithm, and compare experimentally to [3]. In section 9 we define and evaluate the pose search task, based on the presented HPE algorithm. In the last section we conclude and propose possible extensions to our work.

Preliminary versions of several parts of this work were published in [15, 22, 23].

## 2 Background and Related Works

The literature on 2D human pose estimation in still images and videos is vast, and dates back as far as [24, 29]. Both bottom-up [27, 42, 51] and top-down approaches [19, 44] have been proposed. Methods trying to recover the spatial configuration of a human include: matching the entire human shape [25, 45], assembling poses from segmentations [51], relying on detected skin color regions [27, 42], and exploiting contours/gradients to model the shape of body parts [3, 36, 49]. Both exact and approximate inference schemes have been proposed depending on the model structure. For exact inference, tree structured graphs have been extensively used [19, 49, 52], whereas for approximate inference, MCMC sampling [42, 51], loopy belief propagation [57], linear programming [30], or integer quadratic programming [51] algorithms have been employed.

In this section we focus our attention on works most similar in spirit to ours, i.e. those based on pictorial structures. We give special attention to the human parsing technique of Ramanan [49], on which we build directly. Early applications of PS succeeded for naked humans on uncluttered backgrounds [29]. For more challenging images with natural backgrounds and people in unknown clothing it is important to have good appearance models. Many previous works have put great care in estimating them [9, 19, 49, 50]. The most reliable way, but the least automatic, is to derive the appearance models from *manually segmented* parts in a few video frames [9]. Another approach is to apply *background subtraction*, and use the number of foreground pixels at a given position as a unary potential [19, 37, 38]. The *strike-a-pose* work [50] searches all frames for a predefined characteristic pose, easier to detect than a general pose. In this pose all parts are visible and don't overlap, enabling the learning of good appearance models, which are then used to estimate pose in all other frames (assuming stable part appearance over time).

The above strategies cannot be applied to a single image as they require video. The best known automatic method for obtaining person-specific appearance models from a single image, without prior knowledge of the background, is the one of Ramanan [49], described in detail in sec. 2.2. It first tries an initial pose estimation using only generic features, i.e. edges, and then repeat the process with appearance models built on the pose from the initial estimation (fig. 2). This *image parsing* technique was a big advance towards estimating the pose of people with unknown appearance (clothing, poses) from a single image. Recent advances in HPE include using adaptive pose priors [54] or sophisticated image features for detecting body parts, based on gradients [3, 33, 36, 59] or color segmentation [32].

Our pose search application is related to action recognition. The methods we develop are *human-centric*, as we

explicitly detect people in the image and represent the spatial configuration of their body parts. This is complementary to recent works on recognizing actions using low level spatio-temporal features [7, 14, 40, 41, 46] (e.g. shapes from silhouettes [7], histograms of oriented gradients and optical flow extracted densely [40], interest points [41, 46]). There also exist a few works operating at an intermediate level, i.e. detecting people and then describing their action with low-level spatio-temporal features [18]. Our human-centric representations can provide the starting point for action recognition using 2D silhouettes [28] or motion history images [8].

As mentioned above, our approach is based on the image parsing algorithm of [49] which employs the Pictorial Structures framework introduced in [19]. In the rest of this section we review both concepts.

### 2.1 Pictorial Structure Model

We briefly review here the general framework of pictorial structures for human pose estimation.

A person's body parts are represented by a conditional random field. Typically, parts $l_i$ are rectangular image patches and their position is parametrized by location $(x,y)$, orientation $\theta$, scale $s$, and sometimes foreshortening [9, 19]. The posterior of a configuration of parts $L = \{l_i\}$ given an image $I$ is

$$P(L|I,\Theta) \propto \exp\left(\sum_{(i,j)\in E} \Psi(l_i,l_j) + \sum_i \Phi(I|l_i,\Theta)\right) \quad (1)$$

The unary potential $\Phi(I|l_i,\Theta)$ corresponds to the local image evidence for a part in a particular position (likelihood). It depends on appearance models $\Theta$ describing how parts should look like. It computes the dissimilarity between the image patch at $l_i$ and the appearance model for part $i$. The appearance models are parameters of the Pictorial Structures and must be provided by an external mechanism.

The pairwise potential $\Psi(l_i,l_j)$ corresponds to a prior on the relative location of parts. It embeds kinematic constraints (e.g. the upper arms must be attached to the torso) and, in a few works, other relations such as a smooth contour connection between parts [55], occlusion constraints [57] or coordination between physically unconnected parts [37] (e.g. left leg and right arm of a walking person). In many works the model structure $E$ is a tree [19, 22, 49, 50, 52], which enables efficient exact inference, though some works have explored more complex topologies [5, 9, 37, 57, 58, 62, 63, 66] or even fully connected models [64].

Inference returns the single most probable configuration $L^*$ [5, 19], or posterior marginal distributions over the position of each part [22, 49].

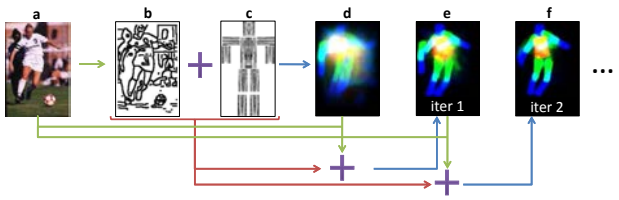



**Fig. 2 Ramanan's image parsing algorithm.** a) input image, b) edges, c) edge templates, d) initial pose soft-estimates after using generic edge-template models only, e-f) refined pose soft-estimates after using both generic and person-specific models

### 2.2 Image Parsing [49]

In Ramanan's work [49], body parts $l_i$ are oriented patches of fixed size, with position parametrized by location $(x,y)$ and orientation $\theta$. They are tied together using eq. (1) into a tree structure $E$ with edges $\Psi(l_i,l_j)$ carrying kinematic constraints of the form

$$\psi(l_i,l_j) = \alpha_i^T \mathrm{bin}(l_i - l_j) \quad (2)$$

where $\mathrm{bin}(\cdot)$ is a vectorized count of spatial and angular histogram bins and $l_i - l_j$ is the relative distance between part $i$ and part $j$ in the coordinate system of the joint between them (hence actually redefining the '-' operator locally for this sentence). Here $\alpha_i$ is a model parameter that favors certain (relative) spatial and angular bins for part $i$ with respect to its parent. This enables capturing more complex distributions than Gaussian priors [3, 19]. The $\alpha_i$ are used for the relative orientation component of $\Psi$ which is an arbitrary multinomial distribution, whereas the relative position $(x,y)$ component of $\Psi$ is a truncated cost that has a uniform value close to the joint location and $+\infty$ everywhere else. In section 7 we will detail how to perform efficient inference in this model.

Since the parts' appearances are initially unknown, Ramanan [49] proposes an iterative *Image parsing* procedure (fig. 2). In a first iteration, the unary potential $\Phi$ only considers image edges, with part templates that are person-independent. After inference, a soft-segmentation for each body part is obtained from the resulting marginal distribution over the part position, by convolving it with a rectangle representing the body part. Part appearance models represented by color histograms are then derived from the soft-segmentations. Finally, inference is repeated with an extended $\Phi$ which includes both person-independent edge templates and the newly acquired color models, which are specific to this particular person and image.

In this scheme, the first inference stage is the mechanism to obtain appearance models. Unfortunately, the edge-based model is not specific enough and the first inference stage typically fails in the presence of somewhat cluttered background leading to poor appearance models and, eventually, incorrect pose estimation.

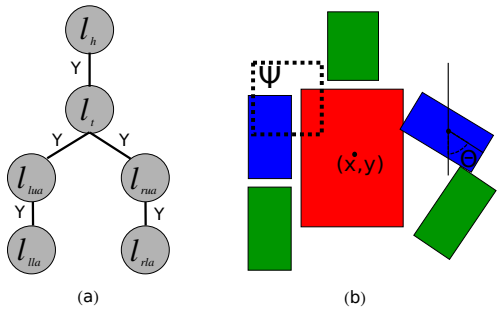



**Fig. 3 Pictorial Structures models.** Each node represents a body part (h: head, t: torso, left/right upper/lower arms lua, rua, lla, rla). (a) The kinematic tree includes edges between every two body parts which are physically connected in the human body. (b) Cardboard representation where body parts are rectangular patches, parametrized by location (x,y) and orientation $\theta$, connected by kinematic priors $\Psi$.

## 3 The Human Upper Body Model in Overview

In this section we describe our model and introduce the key theme of this work: how we can benefit from a generic upper body detection for restricting the position and appearance of the body parts of a person in a particular image. We also give an overview of the algorithm to fit the model to a test image (inference). The following sections then elaborate on the fitting and on how the model is learnt at the various levels.

### 3.1 Benefiting from an upper-body detection

The general idea behind our approach to HPE is to exploit the fact that in the vast majority of amateur photos, movies or TV shows, people appear roughly *upright*, i.e. their head is above their torso. This underpins the design of the helpful preprocessing stages such as upper-body detection (sec. 4) and foreground highlighting (sec. 5). The former finds an approximate position of people in the image and the latter removes background clutter around them. Moreover, we develop orientation priors which naturally emerge out of the head-above-torso assumption (see below) and even derive person-specific appearance models out of part segmentation priors learnt with respect to the detection window (sec. 6). All these innovations progressively reduce the search space for body parts and greatly facilitate the task of Pictorial Structures inference.

### 3.2 Model description

Our upper body Pictorial Structures model consists of 6 body parts, namely torso, head, upper and lower arms connected in a tree structure by the kinematic priors $\Psi(l_i, l_j)$ (fig. 3a). We base the model on [49] (fig. 3b) and extend it with orientation priors described next in this section. We also reduce the spatial extent of the kinematic prior $\Psi$, to specialize it for near-frontal and near-rear views.

***Orientation priors.*** Here, we show how the *upright* assumption can be directly exploited inside the Pictorial Structures model (sec. 2.1).

We extend the model (1) by adding priors $\Upsilon(l_{head}), \Upsilon(l_{torso})$ requiring the orientation of the torso and head to be near-vertical:

$$P(L|I) \propto \exp\left( \sum_{(i,j)\in E} \Psi(l_i, l_j) + \sum_i \Phi(l_i) + \Upsilon(l_{head}) + \Upsilon(l_{torso}) \right) \tag{3}$$

$\Upsilon(\cdot)$ gives uniform probability to a few values of $\theta$ around vertical, and zero probability to other orientations. This reduces the search space for torso and head, thus improving the chances that they will be correctly estimated. Moreover, it also benefits the pose estimation for the arms, because the torso induces constraints on their position through the kinematic prior $\Psi$.

### 3.3 Fitting the model

Here we describe how the model is fitted to novel test images. This process can be summarized in the following stages (fig. 4):

***1. Human detection and tracking.*** We start by detecting human upper-bodies in every image, using a sliding window detector based on the part-based model of [20]. If video frames are processed, then we associate detections over time and each resulting track connects the detections of a different person in every shot.

Detections carry information about the rough position and scale of people in the image. This reduces the search space by setting bounds on the possible (x,y) locations of the body parts and by fixing their scale, thus removing a dimension of the Pictorial Structures' state space entirely. In practice, for each detected person the state space is reduced to a region of the image around the detection, covering the possible arms extent of the person (fig. 4.1).

***2. Foreground highlighting.*** At this stage the search for body parts is only limited by the maximum extent possible for a human of that scale centered on the detected position. We restrict the search area further by exploiting prior knowledge about the structure of the detection window. Relative to it, some areas are very likely to contain part of the person, whereas other areas are very unlikely. This allows the initialization of a GrabCut segmentation [53], which removes part of the background clutter. This stage further constrains the search space by limiting the $(x, y)$ locations to lie within the foreground area determined by GrabCut (fig. 4.2b).

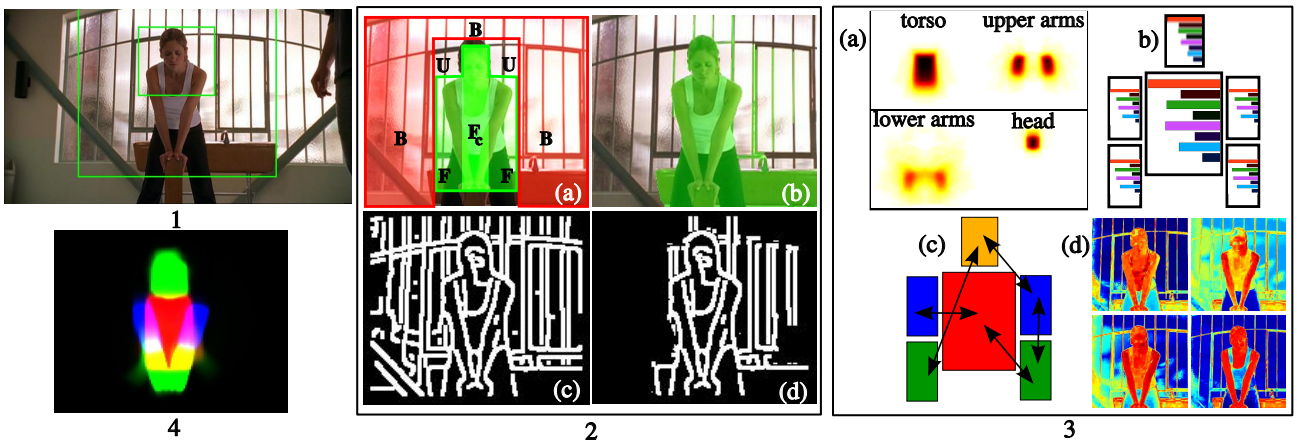



**Fig. 4 Model Fitting. 1) Upper body detection:** The detected person (inner rectangle) and enlarged window where further processing is applied (outer rectangle). **2) Foreground Highlighting:** (a) sub-regions for initializing Grabcut; (b) foreground region output by Grabcut; (c) edges within $\mathscr{R}$; (d) edges remaining after the foreground highlighting stage **3) Estimating appearance models:** (a) Part-specific segmentation priors (SP) applied to $\mathscr{R}$; (b) Initial appearance models obtained from SP (c) appearance models refinement in the appearance transfer stage; (d) Part specific segmentation computed from the refined appearance models (using jet colormap: blue-low, red-high value); clockwise: torso, upper arms, head and lower arms. **4) Inference.** Pose representing fitted model

***3. Estimating appearance models.*** We describe a mechanism for estimating good image and person-specific appearance models from a single image based on two observations: (i) certain body parts have rather stable location w.r.t. the detection window; (ii) often a person's body parts share similar appearance. This mechanism is then used to compute appearance models specific to new instances found in stage 1 (fig. 4.3d).

***4. Parsing.*** An articulated pose is estimated by running inference (eq. (3)) with person-specific appearance models (computed in stage 3) and generic appearance models (edges). The area to be parsed is restricted to the region output of foreground highlighting. Since the person's scale has been fixed by stage 1, no explicit search for body parts over scales is necessary.

The output of the parsing stage is the posterior marginal distribution $P_i(x,y,\theta)$ for every body part of each person detected in the image (fig. 4.4). In the following sections we describe the role of the main components of our approach (sec. 4–7) and evaluate their importance experimentally (sec. 8).

### 3.4 Overview of learning – annotation requirements

Before performing model fitting on new images, several elements of our approach have to be trained (sec. 3.3).

To train the upper body detector used in stage 1 we use images with annotated bounding-boxes around the head and shoulder of humans. The training dataset is described in detail in section 8.1.1.

In order to train the mechanism for estimating person specific appearance models (stage 3), we use images with annotated parts position. For this purpose we use the stickman annotation (fig. 8), i.e. a line segment per body part. The training procedure is described in sections 6.1 and 6.2. The datasets used for training and testing are detailed in section 8.1.2.

For the generic edge templates and kinematic priors within the Pictorial Structures (stage 4) we use the models as trained in [49] and we refer to that paper for further details.

## 4 Upper Body Detection and Tracking

In most shots of movies or TV shows, as well as in many consumer photographs, only the upper body is visible. Here, we train and evaluate a number of upper-body detectors, based on approaches which have previously yielded excellent performance on the related task of rigid object detection [13, 20, 65]. All these detectors use a sliding window mechanism followed by non-maximum suppression.

We start with the approach of [13], where each examined window is subdivided into tiles described by Histogram of Oriented Gradients (HOG) and classified using a linear SVM. Next, we investigate the improvement brought by a part-based hierarchical extension of [13] proposed in [20] (PBM). Finally we check whether complementing an upper-body PBM with a face detector [65] improves performance.

To combine face and upper-body detections, we first transform each face detection to cover the same head-and-shoulder region as an upper-body detection, by regression on the detection window coordinates. The regression parameters are pre-trained to maximize the area of intersection-over-union (IoU) between the regressed windows and real upper-body detections on a separate set of about 10 images. If an upper-body detection and a face detection overlap more than 0.3 in IoU, we discard the latter. This effectively removes double detections of the same person, giving priority to the upper-

body detections, which are typically geometrically more accurate.

In section 8.1.1 we summarize datasets used to train and test the detectors. Then, in section 8.2 we evaluate the detectors and show that the combined face and upper-body detector performs the best among the the proposed ones.

### 4.1 Temporal Association

When video is available we perform an additional temporal association of the detections. After applying the upper-body detector to every frame in the shot independently, we associate the resulting bounding-boxes over time by maximizing their temporal continuity. This produces *tracks*, each connecting detections of the same person.

Temporal association is cast as a grouping problem [61], where the elements to be grouped are bounding-boxes. As similarity measure $s(a,b)$ between two bounding-boxes $a,b$ we use IoU, which subsumes both location and scale information, damped over time:

$$s(a,b) = \mathrm{IoU}(a,b) \cdot e^{-(|a_t - b_t|-1)^2/\sigma^2} \tag{4}$$

where $w_t$ is the frame index where bounding-box $t$ was detected and $\sigma = 2$ controls the rate of temporal damping.

We group detections based on these similarities using the Clique Partitioning algorithm of [21], under the constraint that no two detections from the same frame can be grouped. Essentially, this forms groups maximizing the IoU between nearby time frames.

This algorithm is very rapid, taking less than a second per shot, and is robust to missed detections, because a high IoU attracts bounding-boxes even across a gap of several frames. Moreover, the procedure allows people to overlap partially or to pass in front of each other, because IoU injects a preference for *continuity scale* in the grouping process, in addition to location, which acts as a disambiguation factor. This because the IoU of two bounding-boxes with similar location but different scales is low.

In general, the 'detect & associate' paradigm is substantially more robust than regular tracking, as recently demonstrated by several authors [48, 61].

The temporal association mechanism allows us to further reduce the number of false positives produced by the detector by filtering out short tracks lasting for less than half a second.

## 5 Foreground highlighting

The location and scale information delivered by an upper-body detection greatly constrains the space of possible body parts. They are now confined to the image area surrounding the detection, and their approximate size is known, as proportional to the detection's scale. However, to accommodate for all possible arm poses we must still explore a sizable area (fig. 4.1). Stretching out the arms in any direction forms a large circle centered between the shoulders. In challenging images from TV shows, this area can be highly cluttered, confusing the body part estimator.

Fortunately, we have *prior knowledge* about the structure of the search area. The head lies somewhere in the middle upper-half of the detection window, and the torso is directly underneath it (fig. 4.1). In contrast the arms could be anywhere. This is known because the detector has been explicitly trained to respond to such structures. We exploit this knowledge to initialize Grabcut [53], by learning initial foreground/background color models from regions where the person is likely to be present/absent. The resulting segmentation removes much of the background clutter, substantially simplifying the later search for body parts (fig. 4.2b).

Let $\mathscr{R}$ be a region of interest obtained by enlarging the detection window as in figure 4.1. $\mathscr{R}$ is divided into four sub-regions $F, F_c, B, U$ (see figure 4.2a). Grabcut is initialized as follows: the foreground model is learnt from $F$ and $F_c$ ($F_c$ is known to belong to the person, while $F$ contains mostly foreground, but some background as well); and the background model from $B$ (it covers mostly background, but it might also include part of the arms, depending on the pose). Furthermore, the region $F_c$ is clamped as foreground, but Grabcut is free to set pixel labels in all other sub-regions (we have extended the original Grabcut algorithm to enable these operations). The $U$ region is neutral and no color model is learnt from it. The setup accurately expresses our prior knowledge and results in a controlled, upper-body-specific segmentation, assisted by as much information as we can derive from the previous object detection process. Near the head, $B$ and $F_c$ compete for the $U$ region, with the foreground growing outwards until it meets a background-colored area, resulting in a good head segmentation. Along the sides, the background floods into the initial $F$ to segment the shoulders, while at the same time the arms get labeled as foreground because they are colored more similarly to the initial $F$ than to the initial $B$ (fig. 5).

## 6 Estimating Appearance Models

Employing a person detector greatly reduced the pose search space by fixing the scale dimension, setting bounds on $(x,y)$ locations, and as an initialization in the foreground highlighting stage. Here we propose an even more sophisticated application of the initial detection stage – estimating good person-specific part appearance models from a single image before running the Pictorial Structures inference.

Our approach is motivated by two main observations: (i) the location of some parts relative to the detection win-

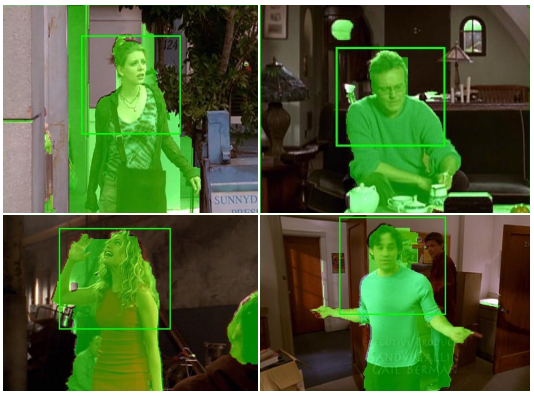


**Fig. 5 Examples of foreground highlighting.** Green overlays depict foreground segmentation selected by our foreground highlighting algorithm. In the presented examples those segmentations include all the body parts and discard the majority of background clutter in the enlarged detection window area.

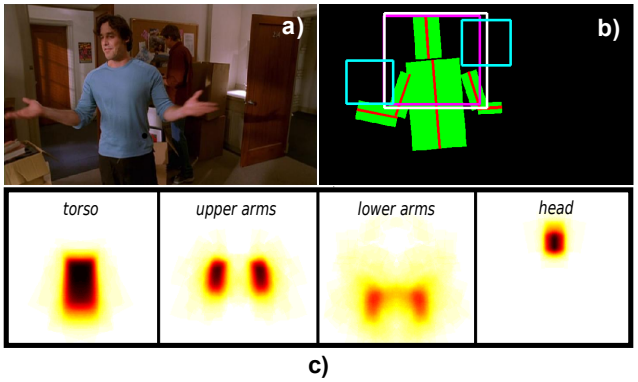



**Fig. 6 Learning segmentation priors.** (a) A training image. (b) Detection windows (cyan); ground-truth stickman (red); expected window (magenta) for this stickman, derived from head and torso sticks together with detector's aspect ratio; detection window (white) associated to the stickman as it overlaps strongly with the expected window; body part rectangles (green) obtained by widening the sticks. (c) Learnt segmentation priors. By estimating left/right arm parts together we increase the number of training examples (this exploits the appearance similarity of symmetric parts, as done in [22, 31, 49]).

dow $W = (x, y, s)$ is rather stable (e.g. the torso is typically in the middle of an upper-body detection window) while others have much higher variability (e.g. lower arms); (ii) the appearances of different body parts are related (e.g. the upper-arms often have the same color as the torso).

To exploit these observations, we learn the relative location distribution of parts w.r.t $W$ and the dependencies between the appearance of different parts from training data. These relations are exploited to generate appearance models for body parts in a new image. In this fashion, parts which are well localized w.r.t to $W$ (e.g. torso) help determining the appearance model for more mobile parts (e.g. lower arms). If no inter-part dependencies exist, our approach naturally degenerates to estimating each part independently.

As the two observations hold in a statistical sense, we learn (i) a segmentation prior (SP) capturing the distribution of the body part locations relative to $W$ (sec. 6.1); (ii) an appearance transfer mechanism to improve the models derived from the segmentation prior by linearly combining models for different body parts (sec. 6.2). SPs are learnt inside the enlarged detection area where the pose estimation algorithm is applied (fig. 4.1). The training data consists of images with ground-truth pose annotated by a stickman (sec. 3.4)

After learning, our method is ready to estimate appearance models on new, unannotated test images (sec. 6.3). Initial appearance models are estimated given $W$ and the learnt segmentation priors. These models are then refined by the appearance transfer mechanism.

### 6.1 Training: learning segmentation priors

For each body part $i$, we learn a *segmentation prior* $SP_i(x, y) \in [0, 1]$: the prior probability for a pixel $(x, y)$ to be covered by the part, before considering the actual image data (fig. 6a). Importantly, pixel coordinates are relative to the detection window, so that SPs can be employed later on test images. Thanks to SPs, we can estimate initial appearance models before running a pictorial structure inference (as opposed to [49]). As in our implementation appearance models are color histograms $P_i(c|fg)$, they are obtained by weighting pixel contributions by $SP_i(x, y)$ (details in section 7).

We learn SPs from training images with ground-truth pose annotated by a stickman (fig. 6a). We first obtain detection windows by running the detector on these images. Next, we associate stickmen to detection windows as in figure 6b. Based on the detection windows, we now project all training stickmen to a common coordinate frame, where they are roughly aligned in location and scale. This imitates the conditions at test time, so that the learnt SPs account for the inaccuracies of the detector. In this common coordinate frame, the SPs are learnt in maximum likelihood fashion: $SP_i(x, y)$ is the fraction of training images where part $i$ covers pixel $(x, y)$. SPs are estimated for every pixel in the enlarged detection area where pose estimation is applied (fig. 4.1).

Example SPs are presented in figure 6c. SPs for the head and torso are quite sharply localized, while SPs for the arms are more diffuse. Interestingly, the location of lower arms appears very uncertain a priori, matching our expectation that they can move around freely.

Notice how SPs are learned in the coordinate frame obtained by actually running the object detector on the training images, as opposed to deriving ideal detection windows from the stickmen. This procedure delivers realistic SPs, tuned to the behavior we expect at test time, as they already account for the uncertainty in the localization of the detection window.

| | torso | upper arms | lower arms | head |
|---|---|---|---|---|
| torso | 1 | 0.13 | 0.12 | 0 |
| upper arms | 0 | 0.87 | 0.30 | 0 |
| lower arms | 0 | 0 | 0.34 | 0 |
| head | 0 | 0 | 0.24 | 1 |

**Table 1 Learned appearance transfer weights.** Each entry $w_{it}$ denotes the contribution of part $i$ (row) to the appearance model of part $t$ (column).

### 6.2 Training: transferring appearance models between body parts

Given an image of a person with lower arms behind their back, can we predict their color based on the visible body parts? Intuitively, we can, because we know that usually people wear either a rather uniformly colored pullover with long-sleeves, in which case the lower arms are colored like the torso, or wear short sleeves, in which case the lower arms have skin color (the same as the face). While external factors might help our reasoning, such as scene type (e.g. beach vs office) and season (winter vs summer), our ability to predict is rooted in the intrinsic relations between the appearance of different body parts.

Inspired by the power of the above relations, here we learn a transfer mechanism to combine the appearance models of different body parts. The input appearance models are derived from SPs (sec 6.1). The appearance transfer mechanism estimates the new appearance model of a part as a linear combination of the input appearance models of all parts.

***Learning mixing weights.*** The new appearance model $AM_i^{TM}$ for a part $t$ is given by

$$AM_t^{TM} = \sum_i w_{it} AM_i^{SP} \tag{5}$$

where $w_{it}$ is the mixing weight of part $i$, in the combination for part $t$, and $AM^{SP}$ is the initial appearance model (derived from the segmentation prior).

The parameters of the transfer mechanism are the mixing weights $w_{it}$. We learn them by minimizing the squared difference between the appearance models produced by the transfer mechanism ($AM_i^{TM}$) and those derived from the ground-truth stickmen ($AM^{GT}$):

$$\min_{w_t} \quad \sum_s \sum_k \left( \sum_i w_{it} AM_{ski}^{SP} - AM_{skt}^{GT} \right)^2 \tag{6}$$

$$s.t: \qquad 0 \leq w_{it} \leq 1, \qquad \sum_i w_{it} = 1$$

where $i$ runs over all parts, $s$ runs over training samples, and $k$ runs over the components of the appearance model (entries of a color histogram, in our case). Ground truth color histograms are computed over rectangular part masks obtained by widening the line segments of the annotated stickman by a predefined factor (fig. 6b). Since (6) has a quadratic objective function with linear inequality constraints, it is a quadratic program. We can find its global optimum efficiently using standard quadratic programming solvers [47]. The mixing weights $w_t$ are found for each part $t$ separately by solving a new quadratic program (6) for each part.

Table 1 shows the mixing weights learnt based on the segmentation prior of figure 6c. Two interesting observations can be made: (i) for parts that are rather stationary w.r.t. the detection window (torso, head), the refined appearance model is identical to the input model from SP; (ii) mobile parts benefit from the contribution of stationary parts with similar appearance. Upper arms models are improved by appearance transfer from the torso. Lower arms, which have the highest localization uncertainty, get strong contribution from all other parts. This because people tend to either wear uniformly colored clothes with long sleeves (contribution from upper arms and torso), or wear short sleeves (contribution from head, which is also skin-colored). These results confirm our intuition that exploiting relations between the appearance of different body parts leads to better appearance models.

### 6.3 Test: estimating appearance models for a new image

After learning SPs and mixing weights of AT, our method is ready to estimate good appearance models for new test images. For clarity, we explain the procedure here for the case where appearance models are color histograms (as in our case). However, our scheme can be applied for other appearance models as well, such as texture histograms.

***Step 1. Estimate color models.*** The procedure entails three steps. First, the detection window $W$ is transformed to the standard coordinate frame where the SPs were learned from, by cropping the enlarged $W$ out of the image and rescaling it to a fixed size. Second, initial color models are derived from the SPs, by weighting color contributions according to the SP values. Third, the color models are refined by applying appearance transfer as in equation (5), leading to the final color models $P_i(c|fg)$.

***Step 2. Estimate color segmentations.*** The color models estimated above characterize the appearance of the body parts themselves. Following [49], we also estimate here a background model $P_i(c|bg)$ for each body part, derived from the complement of the SP (i.e. $1 - SP_i(x,y)$). The foreground $P_i(c|fg)$ and background $P_i(c|bg)$ models are used to derive the posterior probability for a pixel to belong to a part $i$ (using Bayes theorem, assuming $P_i(fg) = P_i(bg)$)

$$P_i(fg|c) = \frac{P_i(c|fg)}{P_i(c|fg) + P_i(c|bg)} \tag{7}$$

The posterior foreground probabilities are then used to derive a color soft-segmentation of the image for each body

part, which is the cue used in the unary term of the pictorial structure ($\Phi$ in equation (1)) (fig. 4.3d). Note that the SPs are used only in steps 1 and 2 to derive appearance models. They are *not* used to restrict the possible location of the parts during pictorial structure inference.

## 7 Implementation details

Here we summarize important technical details of our approach.

***Unary terms.*** The generic and person specific unary terms (eq. 3) are computed by convolving part templates with an edge image (fig. 4.2b) and a part-specific foreground posterior image (fig. 4.3d, equation (7)) respectively, at all locations and orientations (quantized into 24 values) within the enlarged area $\mathscr{R}$ (fig. 4.1). As part templates, we use the discriminatively trained ones of Ramanan [49]. They are trained on the dataset introduced in [49].

***Efficient Pictorial Structure inference.*** The posterior marginals in a tree model (eq. (1)) can be computed using belief propagation (BP), which has $O(nh^2)$ complexity when realized using dynamic programming. In our case, the number of body parts $n$ is 6 and the number of states $h$ is $|x| \cdot |y| \cdot |\theta|$ in the enlarged area after resizing to a standard scale (typically $|x| \simeq |y| = 150$, $|\theta| = 24$). As shown in [19], for certain parametric pairwise kinematic priors $\Psi$ (e.g. Gaussian) the complexity can be reduced to $O(nh)$ by exploiting efficient distance transforms.

In this paper we adopt the non-parametric kinematic prior $\Psi$ of Ramanan [49] (eq. (2)), which could lead to a slow inference ($O(nh^2)$). In practice though, the relative location $(x,y)$ component of $\Psi$, is a truncated cost corresponding to a uniform probability close to the joint location and zero everywhere else. Exploiting this allows us to perform efficient BP using integral images [12] in time independent of the truncation size, which effectively removes a factor $|x| \cdot |y|$ from the complexity. On the other hand, the relative orientation component of $\Psi$ is a true non-parametric distribution, but BP can still be implemented efficiently by using the Fast Fourier Transform to accelerate convolutions. The overall complexity of belief propagation as in [49] is then $O(n \cdot |x| \cdot |y| \cdot |\theta| \cdot \log|\theta|)$ which is very close to $O(n|x| \cdot |y| \cdot |\theta|)$ for $|\theta| << |x| \simeq |y|$.

***Enlarged detection area.*** As shown in Figure 4.1, pose estimation is carried out in a restricted area around a detection window (sec. 3.3), which increases the speed of the algorithm significantly. On the other hand, this area should be large enough to cover the maximal possible arm extent of the detected person. We learn this extent using the training stickman annotations. The procedure is similar to the one used for learning segmentation priors (SP) (sec. 6.1). A detector is run over training images and detections are associated with the ground-truth stickmen (fig. 6). Next, all stickmen are put in the same common coordinate frame, by normalizing the detection window. Finally, we set the enlarged area to be the smallest rectangle enclosing all training body parts. This way, we account for the localization uncertainty of the detector expected at test time, as when learning SP (sec. 6.1).

***Computation times.*** We give here a breakdown of the runtime of our HPE pipeline (sec. 3.3). The results are averaged over 10 runs on a $720 \times 405$ image using a Intel Core 2 Duo E8400 3GHz. The implementation is a mix of C++ and Matlab code [70, 72]. Human detection takes *3.3 sec.* All further processing stages are repeated independently for each detection: (1) foreground highlighting *2.3 sec.*; (2) estimating appearance models: *0.6 sec.*; (3) parsing: computing unary terms *1.5 sec.*, inference *0.8 sec.* (4) overhead of loading models, image resizing, etc.: *1.4 sec.*
After human detection, the total time for HPE on a person is 6.6 sec. The total time for an image is $3.3 + 6.6P$ sec., with $P$ the number of detections.

## 8 Human Pose Estimation Evaluation

Here we present a comprehensive evaluation of our human pose estimation algorithm (HPE). We start by describing the datasets used for training and testing (sec. 8.1). Then we evaluate individually the following components: (i) the upper-body detector (sec. 8.2); (ii) the foreground highlighting stage (sec. 8.3); (iii) the soft-segmentations derived from the new appearance models (sec. 8.4). Finally, we present an extensive evaluation of the actual HPE performance and analyze the impact of various components of our method (sec. 8.5).

### 8.1 Datasets

Here we summarize datasets used to train and test various components of our system.

#### *8.1.1 For person detection*

The upper-body detectors presented in section 4 are trained on a single set of images from three movies (*Run Lola run, Pretty woman, Groundhog day*), manually annotated with bounding-boxes enclosing upper-bodies. This training set contains 96 images which have been selected to maximize diversity, and include many different persons, with only a few images of each, wearing different clothes and/or in different poses. Figure 7 shows some samples from this dataset.

The detectors' evaluation is carried out on a test set of video frames from 'Buffy: the vampire slayer' (season 5

**Fig. 7 Cropped bounding boxes used to train the upper-body detector.** This training set contains both frontal and back views.

episode 2). This dataset contains 164 frames, of which 85 are negative images (i.e. no frontal upper-bodies are visible) and the remaining ones contain 102 instances of frontal upper-bodies. Our detectors as well as the training and test sets are available online [67, 72].

#### *8.1.2 For pose estimation*

Pose estimation performance is evaluated on video frames from the 'Buffy: the vampire slayer' TV show and images from the PASCAL VOC 2008 challenge [17]. We annotated stickmen on episodes 2–6 of Buffy's season 5, for a total of 748 frames. This data is challenging due to uncontrolled conditions, with very cluttered images, often dark illumination, people appearing at a wide range of scales and wearing clothing of any kind and color. The PASCAL data is even more demanding, as it consists mainly of amateur photographs, featuring diverse illumination conditions, low image quality and higher pose variability. A subset of 549 images is used.

For both the Buffy and the PASCAL data, in each image we annotated one roughly upright, approximately frontal person by a 6 part stickman (head, torso, upper and lower arms). This person is visible at least from the waist up and its 6 upper-body parts are fully visible. We name the datasets *Buffy Stickmen* and *ETHZ PASCAL Stickmen*, respectively.

Annotation examples from both datasets are shown in fig. 8 (top), and the pose variability across the datasets is visualized in fig. 8 (bottom). As can be seen, the ETHZ PASCAL Stickmen test set offers greater pose variability than the Buffy Stickmen test set. However, pose variability is only one dimension of the challenge posed by a dataset. In terms of scale, location, clothing and illumination, both datasets are highly challenging. We have released both datasets online [68, 69]. No images from these pose estimation datasets were used for training the person detector (sec. 4).

Later in this section, we investigate the impact of various components of our method on HPE performance in two datasets: (i) when we test on Buffy episodes 2,5,6 (in total 276 images, later referred to as 'Buffy test set') we train on the ETHZ PASCAL dataset and Buffy episodes 3 and 4; (ii) when testing on the ETHZ PASCAL dataset instead, we train from all 5 Buffy episodes.

Additionally we present here some qualitative results on the *Perona November 2009 Challenge*, which is a set of images captured by Pietro Perona and his coworkers in order to challenge our pose estimator. When testing on this dataset, we train our method on all 5 Buffy episodes and all ETHZ PASCAL images.

### 8.2 Person detector evaluation

Here we evaluate the detectors, described earlier in section 4, on the test set detailed in 8.1.1. Figure 9 shows the comparison (detection rate (DR) versus false positives per image (FPPI)) between two frontal view upper-body (ub) detectors: (i) HOG-based ub detector [22]; and (ii) part-based [20] (PBM) ub detector (new in this paper).

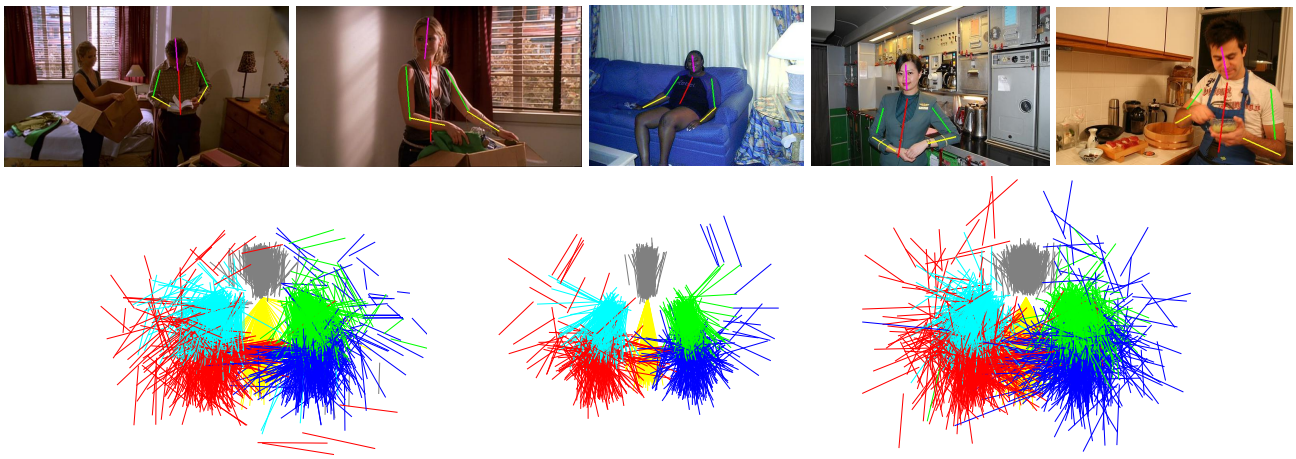
In practice both detectors work well for viewpoints up to 30 degrees away from straight-on frontal and back views. We can see in the plot that the PBM detector improves on the HOG-based ub detector [22] by reducing the number of false positives per image. In particular, if we accept one false positive every ten images (i.e. FPPI $= 0.1$), the detection rate is about 90% (PBM). A detection is counted as correct if its IoU with a ground-truth bounding-box exceeds 0.5, which is the standard PASCAL VOC criterion.

By combining (as described in sec. 4) the PBM ub detector working at an operating point of 0.92 DR / 0.15 FPPI with the face detector of [65] we reach 0.94 DR at 0.43 FPPI. Note that if the operating point of the PBM ub detector were changed to reach the same DR as the combined detector, then its FPPI would increase to 1.57 FPPI. We use the combined face and part-based ub detector (face+PBMub) in the remainder of this paper.

### 8.3 Foreground highlighting evaluation

The foreground highlighting procedure of section 5 is rather conservative and it often retains parts of the background. The goal is not to achieve a perfect segmentation, but to reduce the amount of background clutter (fig. 5). It is more important not to lose body parts, as they cannot be recovered later.

To validate this behavior, we evaluate the obtained foreground segmentations against the annotated stickmen. We count a body part as covered by the foreground segmentation if at least 75% of its stick is covered. On the Buffy test set, foreground highlighting successfully discards 68% of



(a) Buffy Stickmen episodes 3,4 (training subset) (b) Buffy Stickmen episodes 2,5,6 (test subset) (c) ETHZ PASCAL Stickmen (whole dataset = test set)

**Fig. 8 Stickmen datasets.** (top) Example stickmen annotations from the Buffy and ETHZ PASCAL datasets. (bottom) Scatter plots depicting pose variability over a dataset, inspired by [64]. Stickmen are centered on the neck, and scale normalized by the distance between the center of the torso and the top of the head. Hence, the plots capture only pose variability and do not show scale and location variability. The number of annotations in each set are (a) 472, (b) 276, (c) 549. When testing on the Buffy Stickmen test set, we train on the whole ETHZ PASCAL Stickmen and on Buffy episodes 3,4. When testing on ETHZ PASCAL Stickmen instead, we train on all 5 Buffy episodes.

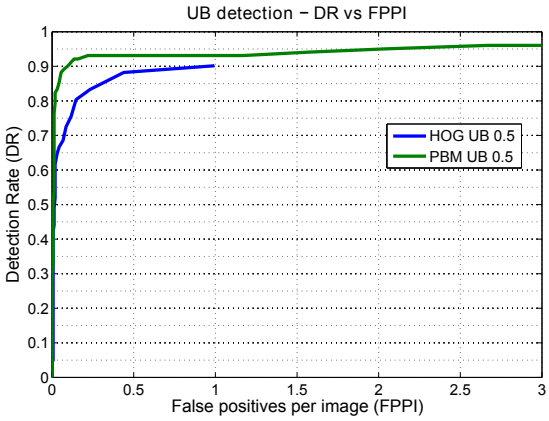



**Fig. 9 Comparison of two frontal upper-body detectors: HOG [13] and PBM [20].** Test on Buffy video frames: 164 frames with 102 positive instances of frontal upper-bodies and 85 negative images. The curves correspond to a correct detection criterion of IoU $> 0.5$. Detection rate is 0.9 at 0.1 FPPI.

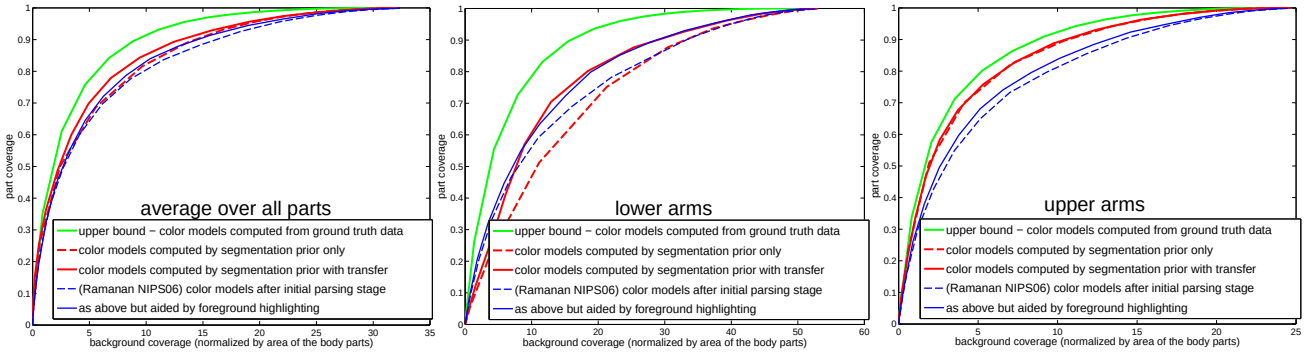
the enlarged detection area while at the same time covering 93.5% of all body parts. This confirms foreground highlighting as an effective preprocessing stage for removing background clutter while preserving the vast majority of the body parts.

In contrast to traditional background subtraction, used in many previous works to extract silhouettes [7, 19, 28], our foreground highlighting method does not need to know the background *a priori*, and allows the background to change over time (in video).

### 8.4 Soft-segmentation evaluation

We compare the quality of part-specific soft-segmentations derived from appearance models generated by several approaches (fig. 10) on the Buffy test set. These segmentations are important for pose estimation, as they form the unary term $\Phi$ of the pictorial structure equation (1). We compare our method to three alternative approaches for estimating color models: (a) edge-based parsing [49]; (b) edge-based parsing aided by foreground highlighting [22]; (c) color models derived from the widened ground-truth stickmen ($AM^{GT}$ in the equation (6)) – this provides an upper bound on the quality of the segmentation that can be achieved with this kind of appearance models. For all approaches, we derive a soft-segmentation from the color models as detailed in section 6.3. All approaches start from detection windows obtained by our upper-body detector (sec. 4)

As figure 10a shows, on average over all body parts, we obtain segmentations on the level of the competing methods already from the initial color models based on segmentation priors (sec. 6.1). Results improve significantly after the appearance transfer stage (sec. 6.2). Interestingly, the color models generated from SPs produce a rather poor segmentation of the lower arms, which have the most diffuse SP (fig. 10b). However, segmentation performance improves substantially after refining the color models by appearance transfer, reaching the performance of the best competing approach. Note, that the competing approaches already involve a Pictorial Structures inference stage. As figure 10c shows, we obtain a considerable improvement over the competitors for upper arms. Arms are especially interesting because they move more than head and torso w.r.t. the detection win-




(a) (b) (c)

**Fig. 10 Evaluation of segmentation induced by color models.** Each curve is averaged over all images in the Buffy test set (episodes 2,5,6). Points on the curve are obtained by thresholding to the soft-segmentation with an increasing threshold. The Y-axis shows how much of the area $A$ of the ground-truth rectangle for a part is covered by the segmentation, in percentage. The X-axis shows how much of the segmentation lies out of the ground-truth rectangle (i.e. on another part or on the background), in multiples of $A$.

dow, making their position harder to estimate, and because they carry most of the semantic pose information necessary to recognize gestures and actions. Importantly, even the ground-truth color models don't lead to perfect segmentation, because the same color might occur on another body part or on the background. On average over all parts, the segmentations derived from our color models are not far from the upper bound (fig. 10a). The largest margin left for improvement is for the lower arms (fig. 10b).

As a side note, figure 10 also shows that foreground highlighting helps [49] finding better appearance models, thus providing a deeper explanation for the improved pose estimation reported in the next section.

### 8.5 Pose estimation evaluation

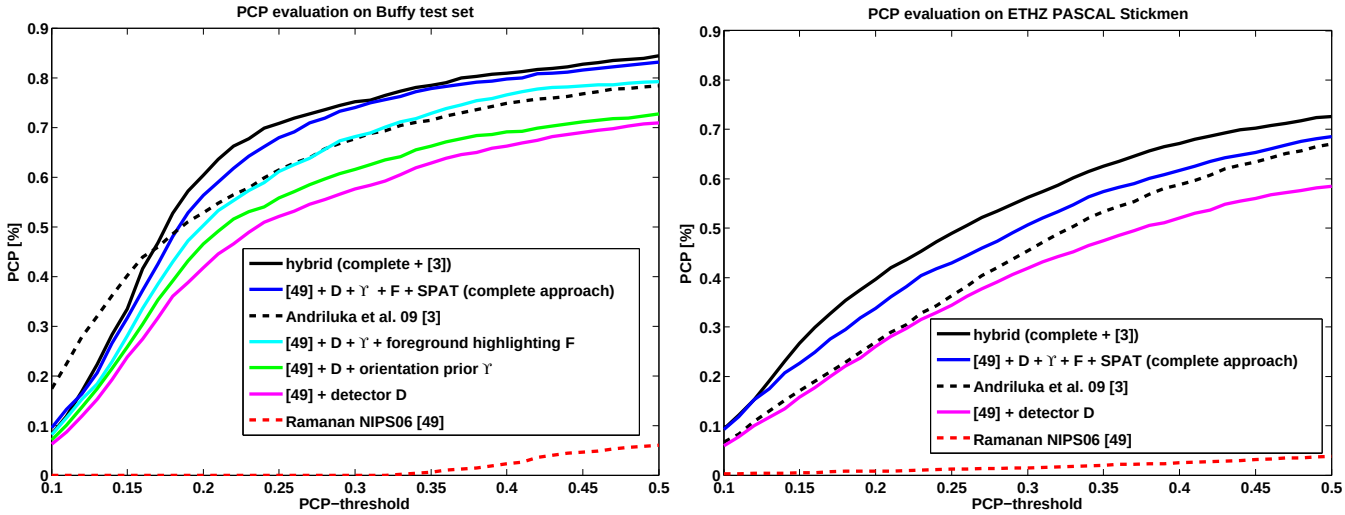
Here, we evaluate how various components of the proposed HPE approach impact the pose estimation performance. We also compare our algorithms with the approach of [3]. The evaluation is carried out for two datasets, i.e. on the Buffy test set or on ETHZ PASCAL test set (details in section 8.1.2).

Performance is measured by *PCP*: the Percentage of Correctly estimated body Parts. An estimated body part is considered correct if its segment endpoints lie within fraction of the length of the ground-truth segment from their annotated location. Varying the fraction (PCP-threshold) between 0.1 and 0.5 we obtain a *PCP-curve* (fig. 11). The lower the PCP-threshold, the stricter the criterion and the more accurate the estimated body parts are deemed correct.

PCP is evaluated only for stickmen that have been correctly localized by the initial upper-body detector (according to the standard IoU 0.5 PASCAL VOC criterion); this is the same criterion used to associate detections to stickmen when learning SPs (fig. 6b). This protocol allows to cleanly evaluate the person detection and pose estimation tasks separately. In all the pose estimation evaluations we used the detector yielding the best performance according to the evaluation from section 8.2, namely the part-based [20] ub detector combined with the face detector of [65] (face+PBMub). The correct localization rate for the Buffy test set and ETHZ PASCAL test set are 95.3% and 75.1% respectively. Therefore, the fact that our approach relies on an initial person detector is not a limiting factor for human pose estimation on the Buffy dataset.

A detailed evaluation of HPE performance for the Buffy test set is shown in figure 11a and summarized in table 2. First, we point out that the original approach of [49] [1] produces very poor results on uncontrolled imagery such as Buffy. Among the reasons for this failure, [49] runs at a fixed scale. Including the scale parameter explicitly into the Pictorial Structures formulation (eq. (1)) could cause vulnerability to local minima and would substantially increase the run-time making the algorithm impractical. The detection preprocessing stage from section 4 allows to estimate poses on images containing people at any scale, while preserving the run-time of [49]. Moreover, it also focuses pose estimation on a small region around the person, removing part of the background clutter and other people in the same image. By running [49] on the scale-normalized enlarged detection area (fig. 4) we already achieve a reasonable performance of 71% (at PCP-threshold 0.5). Incorporating the orientation priors into the Pictorial Structure rises the performance by about 2% (sec. 3.2). Furthermore, removing background clutter using our foreground highlighting procedure brings another 6.5% (sec. 5). Finally, including also the person-specific appearance models (SPAT, sec. 8.4) further boosts PCP by 4.0%. Our complete method outperforms the recent

[1] In all experiments, we use the pictorial structures model of [49] modified by tightened the kinematic prior to specialize it to near frontal and back views, as discussed in section 3.2. This modification, together with the better person detector presented in this paper, are the reasons for the different absolute PCP performance values reported in this paper, compared to the earlier versions [15, 22, 23]. The ranking of the different methods however remained the same.




(a) (b)

**Fig. 11 Pose estimation evaluation - PCP curves:** The performance of the basic framework of [49] is shown as a baseline. The remaining curves show improvements brought by various components of our approach when added on top of [49], namely: person detector (sec. 4); orientation priors (sec. 3.2); foreground highlighting (sec. 5); person-specific appearance model estimation (SPAT, sec. 6). Additionally we show results of the alternative HPE framework proposed by [3] *initialized from our person detector*, and of a hybrid method which uses the body part detectors of [3] inside our complete approach. Results are presented on the Buffy test set (a) and the ETHZ PASCAL stickmen dataset (b).

| average over parts | PCP-threshold | Dataset | [49] | + D (sec. 4) | + D + ϒ (sec. 3.2) | + D + ϒ + F (sec. 5) | + D + ϒ + F + SPAT (sec. 6) (complete approach) | [3] | hybrid |
|---|---|---|---|---|---|---|---|---|---|
| all | 0.5 | Buffy | 6.0% | 71.0% | 72.8% | 79.3% | 83.3% | 78.5% | 84.5% |
| | | PASCAL | 3.8% | 58.5% | — | — | 68.6% | 67.0% | 72.6% |
| | 0.2 | Buffy | 0.0% | 41.8% | 46.6% | 50.3% | 56.4% | 52.9% | 60.4% |
| | | PASCAL | 0.8% | 26.1% | — | — | 33.8% | 26.9% | 39.7% |
| lower arms | 0.5 | Buffy | 3.6% | 47.3% | 47.5% | 57.4% | 61.0% | 50.8% | 60.3% |
| | | PASCAL | 2.7% | 30.0% | — | — | 38.1% | 41.9% | 41.7% |
| | 0.2 | Buffy | 0.0% | 20.5% | 22.8% | 26.4% | 31.8% | 22.4% | 26.4% |
| | | PASCAL | 0.4% | 9.8% | — | — | 12.7% | 12.5% | 12.3% |
| upper arms | 0.5 | Buffy | 6.4% | 67.9% | 71.9% | 81.9% | 89.2% | 87.8% | 93.2% |
| | | PASCAL | 2.9% | 57.2% | — | — | 74.6% | 73.3% | 79.7% |
| | 0.2 | Buffy | 0.0% | 32.1% | 37.3% | 43.5% | 52.9% | 57.4% | 68.6% |
| | | PASCAL | 0.7% | 23.1% | — | — | 34.8% | 35.0% | 43.2% |

**Table 2 Pose estimation evaluation - summary.** Each entry reports a PCP value for a *body-parts-selection*, *PCP-threshold*, *Dataset* and *system-setup* quadruplet. The setups are incremental (i.e. each setup includes all components of the one on its left), except the column labeled [3] which reports the performance of [3].

technique [3] over most of the PCP curve (but not at the strictest PCP-thresholds, below 0.18). For this comparison we used the code released by the authors [2], initialized from the same person detections as our method.

For the ETHZ PASCAL stickmen dataset our approach performs consistently better than the competitors [3, 49] over the whole PCP curve (fig. 11b and table 2). On this dataset all evaluated methods perform worse than on the Buffy test set. This is due to the greater difficulty of the amateur PASCAL images, compared to the professional images from the TV Show Buffy (fig. 8, 12).

[2] We thank Andriluka and Schiele for help in evaluating their approach on our dataset.

The main strength of [3] is in discriminatively trained part detectors based on the shape context descriptor. In a final experiment, we include these part detectors as unary potentials in our complete approach. This hybrid system further improves performance over the whole PCP curve, for both datasets. This confirms the importance of good part detectors [3], and shows that the techniques we propose are nicely complementary to them. In particular, both generic and image specific features are important for HPE.

We can quantify the overall success of a HPE pipeline, including both the person detector and the HPE itself, by the percentage of correctly estimated body parts across all annotated people in a dataset, not only those correctly localized by the person detector. We define the *Total PCP* mea-

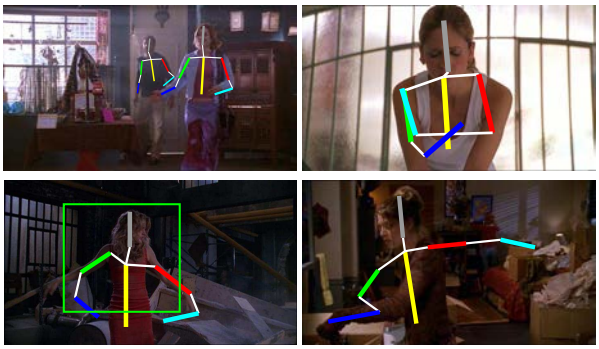


**Fig. 13 HPE failures.** Typical failures occur due to different types of occlusion (top-left: occlusion between people, top-right: body parts out of the image), wrong initial scale estimate (bottom-left) or frontal person detector firing on a strongly side pose (bottom-right);

sure by multiplying PCP by the person localization rate. The *Total PCP* for our complete pipeline are 79.4% and 51.5% on the Buffy and ETHZ PASCAL datasets respectively (at 0.5 PCP-threshold). For the Hybrid approach they are 80.5% and 54.5% respectively.

In figure 12 we present some qualitative examples. Failures may occur due to the lack of occlusion handling, incorrect detection scale or violation of the near frontal/back assumption (when our frontal upper-body detector fires on a strongly side view pose) (fig. 13).

***Repulsive Model*** A well-known problem with Pictorial Structures models is that different body parts can take on similar $(x, y, \theta)$ states, and therefore cover the same image pixels. Typically this happens for the left and right lower arms, when the image likelihood for one is substantially better than the likelihood for the other. It is a consequence of the model being a *tree*, assuming conditional independence between the left and right arms. This is referred to as the *double-counting problem* and has been noted by other authors [19, 57]. One solution, adopted in previous work, is to explicitly model limb occlusion by introducing layers into the model [2, 35, 57], though the graphical model is then no longer a tree.

In order to alleviate the double-counting problem we experimented with a simpler method than layers. We add to the kinematic tree model two *repulsive edges*, connecting the left upper arm to the right upper arm, and the left lower arm to the right lower arm. Again, the model is no longer a tree. These new edges carry a repulsive prior which gives lower probability when two parts overlap than when they don't. Therefore, the extended model *prefers* configurations of body parts where the left and right arms are not superimposed, but it does not forbid them. Approximate inference in the extended model is performed with sum-product Loopy Belief Propagation [6].

Adding the repulsive prior brought a moderate improvement to a stripped down version of our approach when foreground highlighting and estimation of appearance models is not included. However, it did not significantly affect the HPE performance of our complete approach. Therefore, it is not used in the pose search application in the next section. Omitting the repulsive prior keeps the model a tree, which enables efficient exact inference.

**Fig. 14 Pose classes.** Typical poses in the pose classes dataset. From top to bottom: hips, rest and folded.

## 9 Application of HPE – Pose Retrieval

We define *pose retrieval* as the task of retrieving shots in videos containing any person in a given pose from a (possibly large) database of videos (*retrieval database*). Analogous to image retrieval the user can specify the target pose by selecting a single frame containing it. This query frame is not required to belong to the retrieval database, i.e. *external queries* are also supported. A related task was demonstrated in [56], but the query pose was specified by a sketch instead of an image as here. The method employed was very different and involved matching a pattern of self-similarities.

As a second mode of operation, a set of training frames containing the desired pose can be provided, typically covering various people in diverse environments. In this mode, the system has the opportunity to *learn a classifier* specific to the desired pose (see fig. 14 for some typical examples). We refer to the two modes as *query mode* (sec. 9.2), and *classifier mode* (sec. 9.3) respectively.

We investigate the pose retrieval on two parallel threads. The first, the most flexible, is based on the 2D spatial layout of body parts returned by our articulated pose estimation system summarized in section 3.3. Given the spatial layout, we define three pose descriptors and associated similarity measures and compare their performance for pose retrieval (sec. 9.1).

At the end of the section, we explore the second thread, an alternative pose retrieval system based on simpler, lower level features (HOG), which is used as a baseline for comparison with the first thread (sec. 9.4).

**Fig. 12 HPE qualitative results.** rows 1-3: from the ETHZ PASCAL Stickmen dataset; rows 4-6: from the Buffy dataset; row 7: results from the Perona Challenge (note the pose failure in the right-most *Shiva*-like example).

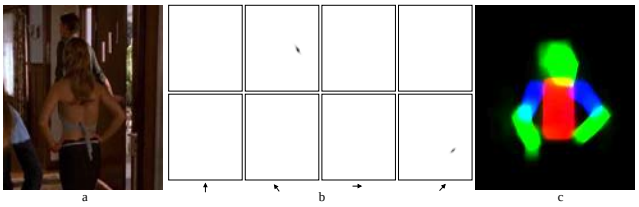



**Fig. 15 Detailed pose estimate.** (a) Input frame (cropped to the enlarged region, as in figure 4.1). (b) Estimated pose for right upper arm (RUA, top) and right lower arm (RLA bottom). Each row shows the posterior marginal $P(l_i = (x,y,\theta))$ as a function of $(x,y)$ for four values of $\theta$ (out of 24). (c) Visualization obtained by convolving rectangles representing body parts, with their corresponding posterior.

## 9.1 Pose descriptors

When video is processed, the procedure in section 3.3 outputs a track of pose estimates for each person in a shot. For each frame in a track, the pose estimate $E = \{E_i\}_{i=1..N}$ consists of the posterior marginal distributions $E_i = P(l_i = (x,y,\theta))$ over the position of each body part $i$ (fig. 4.4), where $N$ is the number of parts. Location $(x,y)$ is in the scale-normalized coordinate frame centered on the person's head delivered by the initial upper body detection, making the representation translation and scale invariant. Moreover, the pose estimation process factors out variations due to clothing and background, making $E$ well suited for pose retrieval, as it conveys a purely spatial arrangements of body parts.

In this section we present three pose descriptors derived from $E$. Of course there is a wide range of descriptors that could be derived and here we only probe three points, varying the dimension of the descriptor and what is represented from $E$. Each one is chosen to emphasize different aspects, e.g. whether absolute position (relative to the original upper body detection) should be used, or only relative (to allow for translation errors in the original detection).

***Descriptor A: part positions.*** A simple descriptor is obtained by downsizing $E$ to make it more compact and robust to small shifts and intra-class variation. Each $E_i$ is initially a $141 \times 159 \times 24$ discrete distribution over $(x,y,\theta)$, and it is resized down separately to $20 \times 16 \times 8$ bins (fig. 16). The overall descriptor $d_A(E)$ is composed of the 6 resized $E_i$, and has $20 \times 16 \times 8 \times 6 = 15360$ values.

***Descriptor B: part orientations, relative locations, and relative orientations.*** The second descriptor encodes the relative locations and relative orientations between pairs of body parts, in addition to absolute orientations of individual body parts.

The probability $P(l_i^o = \theta)$ that part $l_i$ has orientation $\theta$ is obtained by marginalizing out location (fig. 17a)

$$P(l_i^o = \theta) = \sum_{(x,y)} P(l_i = (x,y,\theta)) \tag{8}$$

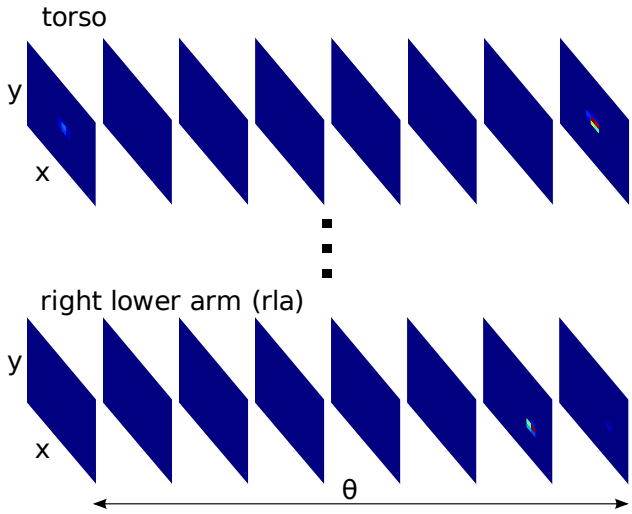



**Fig. 16 Descriptor A.** Obtained by downsizing and concatenating the posterior marginal distributions $E_i$ of all body parts (torso and rla shown, for the example in fig. 15a).

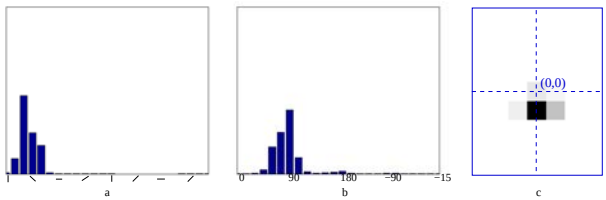



**Fig. 17 Descriptor B.** (a) Distribution over orientations (x-axis) for RUA $P(l_{RUA}^o = \theta)$ from figure 15b. (b) Distribution over relative orientation (x-axis) from RUA to RLA $P(r(l_{RUA}, l_{RLA}) = \rho)$, in degrees. (c) Distribution over relative location (x-axis) from RUA to RLA $P(l_{RLA}^{xy} - l_{RUA}^{xy} = \delta)$.

The probability $P(r(l_i^o, l_j^o) = \rho)$ that the relative orientation $r(l_i^o, l_j^o)$ from part $l_i$ to $l_j$ is $\rho$ is

$$P(r(l_i^o, l_j^o) = \rho) = \sum_{(\theta_i,\theta_j)} P(l_i^o = \theta_i) \cdot P(l_j^o = \theta_j) \cdot \mathbf{1}(r(\theta_i,\theta_j) = \rho) \tag{9}$$

where $r(a,b) = \text{modulo}(a - b, |\theta|)$ is a circular difference operator, and the indicator function $\mathbf{1}(\cdot)$ is 1 when the argument is true, and 0 otherwise. This sums the product of the probabilities of the parts taking on a pair of orientations, over all pairs leading to relative orientation $\rho$ (fig. 17b). It can be implemented efficiently by building a 2D table $T(l_i^o, l_j^o) = P(l_i^o = \theta_i) \cdot P(l_j^o = \theta_j)$ and summing over the diagonals (each diagonal corresponds to a different $\rho$).

The probability $P(l_i^{xy} - l_j^{xy} = \delta)$ of relative location $\delta = (\delta_x, \delta_y)$ is built in an analogous way (fig. 17c). It involves the 4D table $T(l_i^x, l_i^y, l_j^x, l_j^y)$, and summing over lines corresponding to constant $\delta$.

By recording geometric relations between parts, this descriptor can capture local structures characteristic for a pose, such as the right angle between the upper and lower arm in

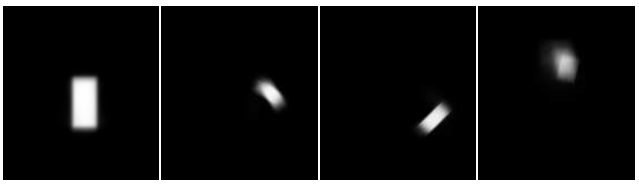


**Fig. 18 Descriptor C.** Soft-segmentations for torso, RUA, RLA and head from figure 15b (displayed here in full resolution; the actual descriptor is downsized).

the 'hips' pose (fig. 14). Moreover, locations of individual parts are not included, only relative locations between parts. This makes the descriptor fully translation invariant, and unaffected by inaccurate initial detections.

To compose the overall descriptor, a distribution over $\theta$ is computed using (8) for each body part, and distributions over $\rho$ and over $\delta$ are computed (9) for each pair of body parts. For the upper-body case, there are 15 pairs and the overall descriptor is the collection of these $6+15+15=36$ distributions. Each orientation distribution, and each relative orientation distribution, has 24 bins. The relative location is downsized to $7\times 9$, resulting in $24\cdot 6+24\cdot 15+9\cdot 7\cdot 15=1449$ total values.

***Descriptor C: part soft-segmentations.*** The third descriptor is based on soft-segmentations. For each body part $l_i$, we derive a soft-segmentation of the image pixels as belonging to $l_i$ or not. This is achieved by convolving a rectangle representing the body part with its corresponding distribution $P(l_i)$. Every pixel in the soft-segmentation takes on a value in $[0,1]$, and can be interpreted as the probability that it belongs to $l_i$ (fig. 18).

Each soft-segmentation is now downsized to $20\times 16$ for compactness and robustness, leading to an overall descriptor of dimensionality $20\times 16\times 6=1920$. As this descriptor captures the silhouette of individual body parts separately, it provides a more distinctive representation of pose compared to a single global silhouette, e.g. as used in [7, 34].

## 9.2 Query mode

In query mode, the user specifies the target pose with a single frame $q$. Through the techniques above, for every person in a shot of the retrieval database we obtain a series of pose descriptors $d_f$, one per video frame $f$ in the track.

In order to search the database for shots containing the target pose, we need (i) a similarity measure between pose descriptors, for comparing the query $d_q$ to descriptors $d_f$ from the database, and (ii) a strategy to score a shot, based on the similarity scores to all the descriptors it contains. The final output of the pose retrieval system is a list of all shots, ranked by their score.

***Similarity measures.*** Each descriptor type (A–C) has an accompanying similarity measure $\mathrm{sim}(d_q,d_f)$:

*Descriptor A.* The combined Bhattacharyya similarity $\rho$ of the descriptor $d^i$ for each body part $i$: $\mathrm{sim}_{\mathrm{A}}(d_q,d_f)=\sum_i \rho(d_q^i,d_f^i)$. As argued in [11], $\rho(a,b)=\sum_j\sqrt{a(j)\cdot b(j)}$ is a suitable measure of the similarity between two discrete distributions $a,b$ (with $j$ running over the histogram bins). *Descriptor B.* The combined Bhattacharyya similarity over all descriptor components: orientation for each body part, relative orientation and relative location for each pair of body parts.
*Descriptor C.* The sum over the similarity of the soft-segmentations $d^i$ for each part: $\mathrm{sim}_{\mathrm{C}}(d_q,d_f)=\sum_i d_q^i\cdot d_f^i$. The dot-product $\cdot$ computes the overlap area between two soft-segmentations, and therefore is a suitable similarity measure.

***Shot scores.*** The score of a shot is set to that of the best scoring track, i.e. the person considered most likely to be carrying out the query pose. We propose here different strategies for scoring a track:
*One-to-one.* The track score is simply the maximum similarity of $d_q$ to every frame: $\max_i \mathrm{sim}(d_q,d_i)$.
*Top-k average.* The track score is the average over the top $k$ frames most similar to $d_q$.
*Query interval.* Consider a short time interval around the query frame $q$. The score of a track frame is the maximum similarity over this query interval. This improves results when pose estimation performs better in a frame near $q$.

The last two strategies can be combined, resulting in a track score integrating several query frames *and* several track frames.

## 9.3 Classifier mode

In classifier mode, a set $\mathscr{S}^+$ of training frames is made available to the system. $\mathscr{S}^+$ includes all frames containing the target pose, from a small number of videos $V$ (e.g. from examples of that pose from a number of shots covering different people and clothing). For frames containing multiple people, $\mathscr{S}^+$ also indicates *which* of them is performing the target pose. A discriminative classifier specific to the desired pose is first learnt, and then used for scoring shots from the retrieval database.

***Training a classifier.*** A linear SVM is trained from $\mathscr{S}^+$ and a negative training set $\mathscr{S}^-$ of frames not containing the target pose. $\mathscr{S}^-$ is constructed by randomly sampling frames from $V$, and then removing those in $\mathscr{S}^+$. The descriptors presented in subsection 9.1 are extracted for all frames in $\mathscr{S}^+$ and $\mathscr{S}^-$, and presented as feature vectors to the SVM trainer. For a frame of $\mathscr{S}^+$, only the descriptor corresponding to the person performing the pose is included.

Optionally, $\mathscr{S}^+$ can be augmented by perturbing the original pose estimates $E$ with small translations and scalings before computing their descriptors. As noted by [39], this practice improves the generalization ability of the classifier. The augmented $\mathscr{S}^+$ is 7 times larger.

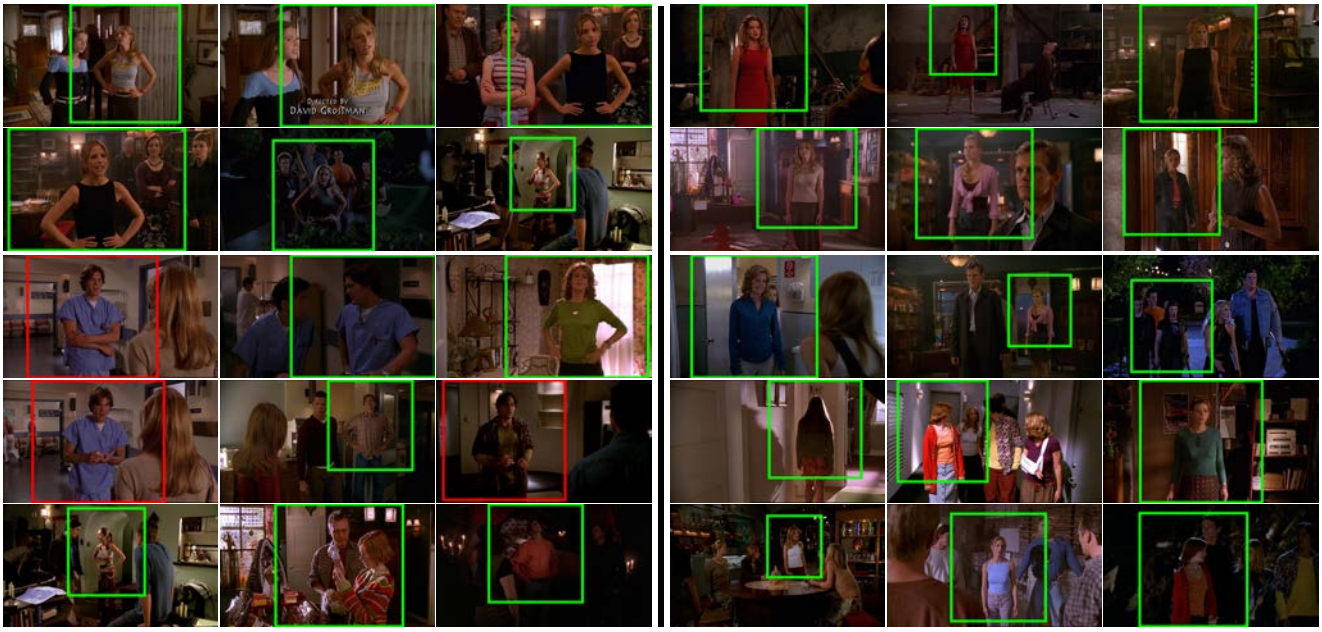


**Fig. 19 Query mode. Left: Hips.** Top 15 returned shots for the result with the highest AP (43.1). The system also returns a box around the person with pose most similar to the query (marked green when correct, and red otherwise). The query is the first frame (Buffy). Among these top few returns there are instances of Buffy wearing different clothes (ranks 3, 4, 6, 13) as well as entirely different characters such as Harmony, Joyce, Riley, and even a vampire (ranks 5, 9, 11, 15 respectively). Notice the large variability in background and lighting conditions. **Right: Rest.** Top 15 returned shots for the result with the highest AP (61.3). Again, the query is the first frame. Note the variety of clothing, backgrounds, and people retrieved starting from a single query frame.

***Searching the database.*** When searching the database the SVM classifier is applied to all descriptors, and the output distance to the hyperplane is used as a score. Therefore, the SVM plays the same role as the similarity measure in query mode. Apart from this, the system operates as in query mode, including using the *top-k average* shot scoring strategy (but not the *query interval* strategy as classifier mode has no query). The classifier mode has the potential to be more accurate than query mode, as it explicitly learns to distinguish the target pose from others. As an additional benefit, the linear SVM can learn which of the components of the feature vector are important from the hyperplane weighting.

### 9.4 Baseline – Hog-based Pose Retrieval

We describe now our baseline pose retrieval system, which uses a Histograms of Oriented Gradients (HOG) [13] descriptor for each upper body detection in a track, rather than the pose descriptors computed from the pictorial structure inference. In order to be able to capture the pose at all in a descriptor, the window must be enlarged over the size of the original upper body detection, and we use here the enlarged region show in figure 4.1 (the same region is used as the starting point for fitting the articulated model). For the HOG computation this is resized to a standard $116 \times 130$ pixels (width $\times$ height).

We employ the HOG pose descriptor for pose retrieval in the same manner as the descriptors of section 9.1:

***Query mode.*** The HOG-based query mode proceeds as in section 9.2, using the negative Euclidean distance between two HOG descriptors as a similarity measure. Other than scale and translation invariance we do not expect this descriptor to have the same invariances as the articulated model descriptors (such as clothing invariance). In particular, we expect it to be very sensitive to background clutter since every gradient in the enlarged region counts.

***Classifier mode.*** Here a classifier is trained for specific poses, e.g. hips, using the same training data as in section 9.3. This has a similar objective to the keyframe pose search of [40] (e.g. a classifier for the pose of coffee at the mouth). As in [13], we use a round of bootstrapping to improve the performance. The classifier from the first round is applied to a large set of negative frames from the training videos (constructed as $\mathscr{S}^-$ in section 9.3). In the second round we add to the negative set the most positively scoring negative frames, so as to double its size, and the classifier is then retrained.

We would expect the classifier to learn to suppress background clutter to some extent, so that this mode would have superior performance over the query mode.

### 9.5 Evaluation of Pose Retrieval

We present experiments on a video database consisting of TV show episodes and Hollywood movies. For each video

the following steps are carried out: first it is partitioned into shots; then our best person detector (face+PBMub) is run on every frame and tracked through the shot (sec. 4, 8.2); for each track, we apply the complete pose estimation algorithm from section 8.5 on every detection; and finally for each detection we have three descriptors (A–C) computed from the fitted articulated model (sec. 9.1), and a HOG descriptor of the enlarged region (which is used for the baseline comparisons, section 9.4).

***Video data and ground truth labelling.*** We show quantitative evaluations on five episodes of the TV series 'Buffy the Vampire Slayer' (episodes 2–6 of the fifth season, a total of 1394 shots containing any upper body, or about 130000 frames). In addition, we also show retrieval examples on five Hollywood movies, 'Gandhi', 'Four Weddings and a Funeral', 'Love Actually', 'About a Boy', 'Notting Hill' for a total of 1960 shots with upper bodies (about 316000 frames).

For the five Buffy episodes every shot is ground truth labelled as to which of three canonical poses it contains: hips, rest, and folded (fig. 14). Three labels are possible indicating whether the shot contains the pose, does not contain the pose, of if the frame is ambiguous for that pose. Ambiguous cases, e.g. when one arm is occluded or outside the image, are ignored in both training and testing. The statistics for these poses are given in table 3. As the ground truth labelling of these episodes is algorithm independent, we use it to assess precision/recall performance for the target poses, and to compare different descriptors and search options. We have released this ground truth annotation online [71].

### *9.5.1 Query Mode - Buffy*

For each pose we select 7 query frames from the 5 Buffy episodes. Having several queries for each pose allows to average out performance variations due to different queries, leading to more stable quantitative evaluations. Each query is searched for in all 5 episodes, which form the retrieval database for this experiment. For each query, performance is assessed by the average precision (AP), which is the area under the precision/recall curve. As a summary measure for each pose, we compute the mean AP over its 7 queries (mAP). Four queries for each pose are shown in figure 14. In all quantitative evaluations, we run the search over all shots containing at least one upper body track.

***Shot scores.*** We investigate the impact of the different strategies for scoring tracks, while keeping the descriptor fixed to A (sec. 9.2). Both ideas of *query interval* and *top-k average* bring a moderate improvement. We found a query interval of 5 frames and $k = 10$ to perform best overall, e.g.it improves mAP for 'rest' to 52.8%, from the 49.3% achieved by the straightforward *one-to-one* approach. In the following experiments, we leave these parameters fixed at these values.

| | A | B | C | HOG | instances | chance |
|---|---|---|---|---|---|---|
| hips | 24.8 | **32.5** | 22.0 | 8.9 | 31 / 983 | 3.2 % |
| rest | 47.3 | **52.8** | 47.2 | 20.3 | 108 / 950 | 11.4 % |
| folded | **16.8** | 16.2 | 14.5 | 8.9 | 49 / 991 | 4.9 % |

**Table 3 Experiment 1.** Query mode (test set = episodes 2–6). For each pose and descriptor, the table reports the mean average precision (mAP) over 7 query frames. The 'instances' column shows the number of instances of the pose in the database, versus the total number of shots searched (the number of shot varies due to different poses having different numbers of shots marked as ambiguous in the ground-truth). The 'chance' column shows the corresponding chance level.

***Descriptors.*** As table 3 shows, pose retrieval based on articulated pose estimation performs substantially better than the HOG baseline (sec. 9.4), on all poses, and for all three descriptors we propose (sec. 9.1). As the query pose occurs infrequently in the database, absolute performance is far above chance (e.g. 'hips' occurs only in 3% of the shots), and we consider it very good given the high challenge posed by the task [3]. Notice how HOG also performs better than chance, because shots with frames very similar to the query are highly ranked, but it fails to generalize.

As shown in figure 19, our method succeeds in returning different people, wearing different clothes, at various scales, background, and lighting conditions, starting from a *single* query frame. Interestingly, the complex descriptor B performs best on average, which shows the benefits of capturing the geometry of poses as completely as possible. Moreover, the simpler soft-segmentation descriptor C performs worst (among the three we propose).

### *9.5.2 Classifier Mode - Buffy*

We evaluate here the classifier mode. For each pose we use episodes 2 and 3 as the set $V$ used to train the classifier (sec. 9.3). The positive training set $\mathscr{S}^+$ contains all time intervals over which a person holds the pose (also marked in the ground-truth). The classifier is then tested on the remaining episodes (4,5,6). Again we assess performance using mAP. In order to compare fairly to query mode, for each pose we re-run using only query frames from episodes 2 and 3 and searching only on episodes 4–6 (there are 3 such queries for hips, 3 for rest, and 2 for folded). Results are given in table 4, which report averages over 3 runs (as the negative training samples $\mathscr{S}^-$ are randomly sampled).

Several interesting observations can be made. First, the three articulated pose descriptors A–C do substantially better than HOG on hips and rest also in classifier mode. This highlights their suitability for pose retrieval. On folded, for which very few training examples are available, descriptor A performs close to HOG. Second, when compared on the

[3] The pose retrieval task is harder than simply classifying images into three pose classes. For each query the entire database of 5 full-length episodes is searched, which contains many different poses.

| | Classifier Mode | | | | Query mode | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | HOG | A | B | C | HOG |
| hips | 27.7 | **30.9** | 12.6 | 5.3 | 19.6 | 27.5 | 16.4 | 2.1 |
| rest | 51.0 | **54.6** | 46.5 | 24.6 | 45.2 | 47.6 | 48.4 | 14.9 |
| folded | **11.1** | 8.6 | 8.7 | 13.3 | 10.8 | 12.0 | 12.4 | 5.9 |

**Table 4 Experiment 2.** Left columns: classifier mode (test set = episodes 4–6). Right columns: query mode on same test episodes 4–6 and using only queries from episodes 2 and 3. Each entry reports AP for a different combination of pose and descriptor, averaged over 3 runs (as the negative training samples $\mathscr{S}^-$ are randomly sampled).

| Pose | *Exp-A* | *Exp-B* | *Exp-C* |
|---|---|---|---|
| hips | 21.1 | **29.9** | 27.5 |
| rest | 34.7 | 58.3 | **62.8** |
| folded | 12.8 | **28.7** | 16.2 |

**Table 5 Part-based specific pose detection (classifier mode): mAP over 3 runs.** Each column corresponds to a different experimental setup (see main text for details).

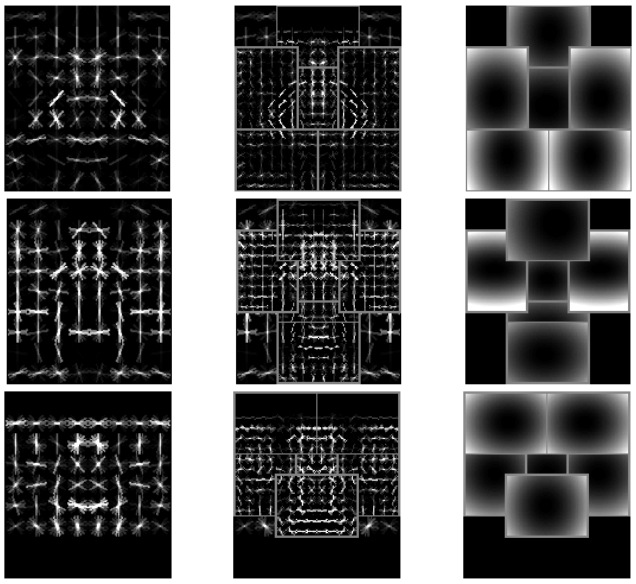


**Fig. 20 Part-based models.** Columns (from left to right): root HOG filter; HOG filters per part; deformation models per part. Rows correspond to pose classes (from top to bottom): *hips*, *rest* and *folded*. Note how the respective pose class is visible in the learned filters.

same test data, HOG performs better in classifier mode than in query mode, for all poses. This confirms our expectations from section 9.4, as it can learn to suppress background clutter and to generalize to other clothing/people, to some extent. Third, the complex articulated pose descriptors A and B, which do well already in query mode, benefit from classifier mode when there is enough training data (i.e. on the hips and rest poses). There are 43 instances of rest in episodes 2 and 3, and 17 of hips, but only 11 of folded. To further complicate the learning task, not all training poses are correctly estimated (see evaluation in section 8.5). This phenomenon is consistent over all three descriptors. As in section 9.5.1, descriptor B performs best overall.

### *9.5.3 Pose specific part-based models (PBM)*

In this experiment, we use the model proposed by Felzenszwalb et al [20], which is the state-of-the art in object category detection [17]. We use this model to train discriminative pose specific detectors for our three pose classes (i.e. *hips*, *rest* and *folded*).

The positive training set comes from the enlarged regions derived from the upper body detections, as in section 9.4. The negative training set, for a given pose class, is composed of samples from the two other pose classes [4] . Additionally, experiment Exp-C also uses additional background images without people (rightmost column of table 5, more details below).

*Exp-A* uses models consisting of just the root filter of [20], whereas models in column *Exp-B* have also six parts each (see right column in fig.20). Both experiments are exactly comparable to the two approaches evaluated in section 9.5.2) since they only evaluate the enlarged upper-body detection windows. In order to push to its limits the idea of doing pose search through a generic object detector, in *Exp-C* we run the pose detectors in sliding-window mode (i.e. evaluating every window instead of only those returned by the upper-body detector). Models for this experiment are trained with additional negative images containing no people (i.e. the INRIA Person negative training dataset [5]) since they must learn to reject image windows without people at test time.

Table 5 shows the results (mAP over 3 runs, as negative training samples are randomly selected). The strategy used to score the shots is the top-$k$ average with $k = 10$ as in query mode (sec. 9.2). The model parts significantly improve performance (from Exp-A to Exp-B, table 5). Moreover, these PBM models perform much better than the HOG ones (table 4, 'classifier mode', HOG column) and about as well as our descriptors based on explicit human pose estimation for hips and rest (table 4 'classifier mode', B column). An important advantage of the proposed descriptors based on human pose estimation over the more direct PBM approach is that they enable pose retrieval given a *single query frame* (sec. 9.2). In fact our pose descriptors perform comparably well in query mode as in classifier mode. This opens the way for realistic video search applications. The PBM approach instead is only possible in classifier mode, where the user must provide several positive training samples of the pose class. Finally, note how *Exp-C* performs slightly below *Exp-B* on average over all pose classes, suggesting that evaluating only upper-body detection windows does not limit the

[4] Number of positive training samples (annotated image windows on ub detections) per pose: 563 hips, 2206 rest and 777 folded. The negative sets (other poses) contain: 2983 for hips, 1340 for rest and 2769 for folded.

[5] `http://pascal.inrialpes.fr/data/human/`

performance of the direct PBM approach. Notice how *Exp-C* could not be used for pose search in practice, as it takes several seconds per video frame.

Figure 20 shows the three learned pose specific models for experiment *Exp-B*. Each row corresponds to a different pose class. The left column shows the root filter, the middle column the six parts overlaid on the root filter, and the right column the spatial layout of the deformation model.

### *9.5.4 Hollywood Movies*

To test the generalization ability of the proposed pose representation even further, we search several Hollywood movies ('Gandhi', 'Four Weddings and a Funeral', 'Love Actually', 'About a Boy', and 'Notting Hill') using several queries from the first three movies (fig. 21). As the figure shows, our method can retrieve a variety of different poses, and finds matches over different people and across different movies.

**Fig. 21 Retrieval on Hollywood movies.** The top few returned shots for each of 3 queries (rank marked on the top left; an image with several ranks indicates a succession of very similar returns). The queries are from 'Gandhi', 'Four Weddings and a Funeral', and 'Love Actually', and the search is over all of those three and 'About a boy', 'Notting Hill'. The first image is the query in each case. Notice the difference in illumination conditions, background, clothing and person between the query and the returned shots. Also, often the system successfully returns correct shots from a different movie than the one the query comes from (e.g. the 5th ranked return in the top example is from 'Gandhi', while the query is from 'Four Weddings and a Funeral'). We have manually marked incorrect returns in red (we do not have ground-truth for these videos).

## 10 Conclusions

We have presented a fully automated 2D articulated human pose estimation able to work with uncontrolled images. As it estimates poses based on a single image, it can be used to process both videos and individual images. It handles people appearing at any scale, in diverse illumination conditions and wearing any type/color clothes. The method works equally well for any skin color (fig. 12, first row). The only assumption we exploit is that people appear upright and are seen from an approximately frontal or back viewpoint. Our HPE approach is not specific to upper-bodies only and can easily be extended to full-body configurations (as already available in our software release [70]).

We showed experimentally that all components of the proposed HPE approach contribute to pose estimation performance and that our complete approach improves over the recent technique [3] as well as the earlier method of [49], on which our method builds. However, after our results were originally published [15], better results were reported by Sapp et al [54]. Their work finds the most similar training images to a test image, and then builds an image specific pose prior from the corresponding stickmen annotations. Their method outperforms both our approach and [3], achieving 85.9 and 79.0 PCP at 0.5 PCP-threshold on the Buffy Stickmen and ETHZ PASCAL Stickmen datasets respectively.

Additionally, we presented a successful application of our HPE technique, called *pose search*. We demonstrated that pose retrieval is possible on video material of high difficulty and variety, starting from a single query frame. This opens up the possibility of further video analysis looking at combinations of poses over time, and over several characters within the same shot (interactions). Analogous pose search methods can also be developed for other (non-human) animals.

***Future work.*** As shown in figure 13, there is still room for improvement in terms of pose estimation, as currently it can neither handle occlusions nor recover from a wrong initial scale estimate. We intend to address these issues in the future. In our recent paper [16] we present a first attempt at dealing with occlusions caused by other nearby people or by the limited extent of the image (but not by occluding objects such as desks or lampposts).

**Acknowledgements** We are grateful to Deva Ramanan for help understanding his code; to Pietro Perona for collecting the Perona Challenge images; to Micha Andriluka and Bernt Schiele for help running their technique on our datasets. This work was partially funded by the European research project CLASS, the Swiss National Science Foun-

dation (SNSF), ERC grant VisRec no. 228180, the Royal Academy of Engineering, Microsoft and Junta de Andalucía.

## Appendix: Released materials

We have released a variety of output from the research that led to this paper: (i) the person detectors together with their training and test sets [67, 72]; (ii) the Buffy Stickmen [68] and ETHZ PASCAL Stickmen [69] datasets together with the matlab code of our HPE evaluation framework and our PCP performance curves; (iii) the complete source code of our HPE technique [70]; (iv) a demo webpage where users can upload their images and get the pose estimation result back [73]; (v) the ground truth annotation for pose search on the Buffy data [71].

## References


1. Agarwal A, Triggs B (2004) 3d human pose from silhouettes by relevance vector regression. In: CVPR
2. Agarwal A, Triggs B (2004) Tracking articulated motion using a mixture of autoregressive models. In: ECCV
3. Andriluka M, Roth S, Schiele B (2009) Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR
4. Arandjelovic O, Zisserman A (2005) Automatic face recognition for film character retrieval in feature-length films. In: CVPR
5. Bergtholdt M, Knappes J, Schnorr C (2008) Learning of graphical models and efficient inference for object class recognition. In: DAGM
6. Bishop C (2006) Pattern recognition and machine learning. Springer
7. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: ICCV
8. Bobick A, Davis J (2001) The recognition of human movement using temporal templates. IEEE Trans on PAMI 23(3):257–267
9. Buehler P, Everinghan M, Huttenlocher D, Zisserman A (2008) Long term arm and hand tracking for continuous sign language tv broadcasts. In: BMVC
10. Cham T, Rehg J (1999) A multiple hypothesis approach to figure tracking. In: CVPR
11. Comaniciu D, Meer P (2002) Mean shift: A robust approach toward feature space analysis. In: IEEE Trans. on PAMI
12. Crow F (1984) Summed-area tables for texture mapping. In: SIGGRAPH
13. Dalal N, Triggs B (2005) Histogram of Oriented Gradients for Human Detection. In: CVPR, vol 2, pp 886–893
14. Dollar P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: ICCV VS-PETS
15. Eichner M, Ferrari V (2009) Better appearance models for pictorial structures. In: BMVC
16. Eichner M, Ferrari V (2010) We are family: Joint pose estimation of multiple persons. In: ECCV
17. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2008) The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html
18. Fathi A, Mori G (2008) Action recognition by learning mid-level motion features. In: CVPR
19. Felzenszwalb P, Huttenlocher D (2005) Pictorial structures for object recognition. IJCV 61(1)
20. Felzenszwalb P, McAllester D, Ramanan D (2008) A discriminatively trained, multiscale, deformable part model. In: CVPR
21. Ferrari V, Tuytelaars T, Van Gool L (2001) Real-time affine region tracking and coplanar grouping. In: CVPR
22. Ferrari V, Marin-Jimenez M, Zisserman A (2008) Progressive search space reduction for human pose estimation. In: CVPR
23. Ferrari V, Marin-Jimenez M, Zisserman A (2009) Pose search: retrieving people using their pose. In: CVPR
24. Forsyth D, Fleck M (1997) Body plans. In: CVPR
25. Gavrilla DM (2000) Pedestrian detection from a moving vehicle. In: ECCV00, vol 2, pp 37–49
26. Guan P, Weiss A, Balan A, Black M (2009) Estimating human shape and pose from a single image. In: ICCV
27. Hua G, Yang MH, Wu Y (2005) Learning to estimate human pose with data driven belief propagation. In: CVPR
28. Ikizler N, Duygulu P (2007) Human action recognition using distribution of oriented rectangular patches. In: ICCV workshop on Human Motion Understanding
29. Ioffe S, Forsyth D (1999) Finding people by sampling. In: ICCV
30. Jiang H (2009) Human pose estimation using consistent max-covering. In: ICCV
31. Jiang H, Martin DR (2008) Global pose estimation using non-tree models. In: CVPR
32. Johnson S, Everingham M (2009) Combining discriminative appearance and segmentation cues for articulated human pose estimation. In: MLVMA
33. Johnson S, Everingham M (2010) Clustered pose and nonlinear appearance models for human pose estimation. In: BMVC
34. Ke Y, Sukthankar R, Hebert M (2007) Spatio-temporal shape and flow correlation for action recognition. In: CVPR

35. Kumar MP, Torr PHS, Zisserman A (2004) Learning layered pictorial structures from video. In: ICVGIP, pp 148–153
36. Kumar MP, Torr PHS, Zisserman A (2009) Efficient discriminative learning of parts-based models. In: ICCV
37. Lan X, Huttenlocher D (2005) Beyond trees: Common-factor models for 2D human pose recovery. In: ICCV, vol 1
38. Lan X, Huttenlocher DP (2004) A unified spatio-temporal articulated model for tracking. In: CVPR, vol 1, pp 722–729
39. Laptev I (2006) Improvements of object detection using boosted histograms. In: BMVC
40. Laptev I, Perez P (2007) Retrieving actions in movies. In: ICCV
41. Laptev I, Marszałek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: CVPR
42. Lee MW, Cohen I (2004) Proposal maps driven mcmc for estimating human body pose in static images. In: CVPR
43. Li P, Ai H, Li Y, Huang C (2007) Video parsing based on head tracking and face recognition. In: CIVR
44. Mikolajczyk K, Schmid C, Zisserman A (2004) Human detection based on a probabilistic assembly of robust part detectors. In: ECCV, Springer-Verlag
45. Mori G, Malik J (2002) Estimating human body configurations using shape context matching. In: CVPR
46. Niebles J, Fei-Fei L (2007) A hierarchical model model of shape and appearance for human action classification. In: CVPR
47. Nocedal J, Wright S (2006) Numerical Optimization, Springer-Verlag
48. Ozuysal M, Lepetit V, Fleuret F, Fua P (2006) Feature harvesting for tracking-by-detection. In: ECCV
49. Ramanan D (2006) Learning to parse images of articulated bodies. In: NIPS
50. Ramanan D, Forsyth DA, Zisserman A (2005) Strike a pose: Tracking people by finding stylized poses. In: CVPR, vol 1, pp 271–278
51. Ren X, Berg A, Malik J (2005) Recovering human body configurations using pairwise constraints between parts. In: CVPR
52. Ronfard R, Schmid C, Triggs B (2002) Learning to parse pictures of people. In: ECCV
53. Rother C, Kolmogorov V, Blake A (2004) Grabcut: interactive foreground extraction using iterated graph cuts. SIGGRAPH 23(3):309–314, DOI http://doi.acm.org/10.1145/1015706.1015720
54. Sapp B, Jordan C, Taskar B (2010) Adaptive pose priors for pictorial structures. In: CVPR
55. Sapp B, Toshev A, Taskar B (2010) Cascaded models for articulated pose estimation. In: ECCV
56. Shechtman E, Irani M (2007) Matching local self-similarities across images and videos. In: CVPR, Minneapolis, MN, USA
57. Sigal L, Black M (2006) Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In: CVPR
58. Sigal L, Isard M, Sigelman BH, Black MJ (2003) Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In: NIPS
59. Singh VK, Nevatia R, Huang C (2010) Efficient inference with multiple heterogeneous part detectors for human pose estimation. In: ECCV
60. Sivic J, Zisserman A (2003) Video Google: A text retrieval approach to object matching in videos. In: ICCV
61. Sivic J, Everingham M, Zisserman A (2005) Person spotting: video shot retrieval for face sets. In: CIVR
62. Tian TP, Sclaroff S (2010) Fast globally optimal 2d human detection with loopy graph models. In: CVPR
63. Tian TP, Sclaroff S (2010) Fast multi-aspect 2d human detection. In: ECCV
64. Tran D, Forsyth D (2010) Improved human parsing with a full relational model. In: ECCV
65. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: CVPR, pp 511–518
66. Wang Y, Mori G (2008) Multiple tree models for occlusion and spatial constraints in human pose estimation. In: ECCV
67. website (2008) VGG Upper body detector. `http://www.robots.ox.ac.uk/~vgg/software/UpperBody/`
68. website (2009) Buffy Stickmen dataset. `http://www.robots.ox.ac.uk/~vgg/data/stickmen/`
69. website (2009) ETHZ PASCAL Stickmen dataset. `http://www.vision.ee.ethz.ch/~calvin/ethz_pascal_stickmen/`
70. website (2009) HPE software. `http://www.vision.ee.ethz.ch/~calvin/articulated_human_pose_estimation_code/`
71. website (2009) VGG pose estimation and search. `http://www.robots.ox.ac.uk/~vgg/research/pose_estimation/`
72. website (2010) CALVIN Upper Body Detector. `http://www.vision.ee.ethz.ch/~calvin/calvin_upperbody_detector/`
73. website (2010) HPE online demo. `http://www.vision.ee.ethz.ch/~hpedemo/`