Human Pose Estimation Using Patch-based Candidate Generation and Model-based Verification

Kota Hara, Takaharu Kurokawa
Intelligent Systems Laboratory, SECOM, Tokyo, Japan
{kot-hara, t-kurokawa}@sec.com.co.jp

Abstract—We present an algorithm for 3d pose estimation of articulated people in natural images. The poses are disassembled into a collection of local patches and a new pose is inferred by assembling the local patches. This concept allows inference of a wide variety of poses from a small number of training patches. The actual process is realized efficiently by a novel voting scheme where each local patch extracted from the input image is matched to the model patches and matchings cast votes for possible locations and poses of the subject, yielding a set of candidate location-pose pairs. Each candidate is then holistically verified using a top-down model based method, where SVM regression computes the final score by aggregating the subscores which capture different features of the candidates. We evaluate our method on both real and synthetic images and demonstrate its ability.

I. INTRODUCTION

Human pose estimation is a key component to various applications such as video surveillance, human computer interfaces and markerless motion capture. Due to the effort made by the research community, there has been significant improvements in the area. However, there are some issues which prevent pose estimation from being used in “real world” environments where complex backgrounds, a wide variety of poses and multiple people are presented.

In this paper, we describe a method for 3d pose estimation from “real world” monocular images with severe background clutter. While some approaches [1], [2], [3], [4], [5], [6], [7], [8] use a silhouette obtained via background subtraction to avoid directly handling background clutter, our method does not rely on background subtraction which is reliable primarily in stable lighting conditions. Furthermore, our method can be used to detect multiple people in a given image and estimate all poses simultaneously. In the case where the input image is assumed to contain only one subject, our approach works as a pose estimation method with robustness to the position of the subject.

Due to the highly articulated nature of the human body, the number of possible poses is enormous, making the direct inference of poses from images difficult. We tackle this problem by modeling the human body as a collection of local patches extracted from various locations of different poses and inferring a new pose by assembling these local patches. This concept allows inference of a wide variety of poses from a small number of training poses. The procedure is realized efficiently through a novel voting scheme, where several local patches extracted from a given input image are matched to the model patches and matchings cast votes for possible locations and poses of the subject, yielding a set of candidate location-pose pairs. Each candidate is then holistically verified using a top-down model based method, where SVM regression computes the final score by aggregating the subscores which capture different features of the candidates. The main flow of our approach is depicted in Fig. 1.

Our approach can be viewed as an extension of the implicit shape model [9], a voting-based object detection method, used in both human detection and pose estimation. A voting-based method can have a good localization ability as it estimates the location of the object’s reference point. To avoid the normalization problem of previous formalizations of the voting-based method, we present a new interpretation and probabilistic formalization of the voting scheme in the context of object detection method based on sliding-window search [10]. We then present a formalization of the voting scheme for joint detection and pose estimation of people along with an efficient mode seeking procedure to find candidates. Other key contributions are a new patch description and its distance measure and a novel top-down candidate verification method. We evaluate our method on both real and synthetic images with severe background clutter and demonstrate its capability in localizing people and inferring their poses.

The paper is organized as follows: Section 2 describes related work. Section 3 describes the candidate generation using local patches. In section 4, the top-down model based verification of candidates is presented. Finally, section 5 shows the experimental results and section 6 concludes the paper.

II. RELATED WORK

Due to space limitations, we give only a brief survey of pose estimation algorithms that do not rely on background subtraction, temporal information or multiple cameras. One class of approaches involves learning a discriminative model which maps image features to pose space [11], [12], [13], [14], [15], [16]. Agarwal and Triggs [11] use local descriptors to estimate human upper body poses in 3D using direct regression. Kanaujia [12] improves tolerance to deformation, misalignment and clutter in training and test sets using a hierarchical image descriptor. Ning et al. [13] propose a discriminative bag-of-words approach using Bayesian mixtures of experts to represent the multi-modal distribution of the 3D human pose space. Rogez et al. [14] first detect and classify
human poses using random forests and then estimate poses by weighting the mean poses of the classes. Similarly, Okada and Soatto [15] classify poses into pose clusters and learn mappings from each cluster to pose space using linear RVM regressors. Generally this class of approaches has difficulty accommodating a wide variety of poses and is sensitive to misalignment (except for [13]).

The other common class of approaches finds the pose which maximizes certain functions using optimization [17], [18], [19], [20], [21], [22], [23]. To reduce the computation cost, some approaches explicitly detect body parts using a body limb or face detector and then find the most plausible configuration of the parts [18], [19], [20], [22]. Although these approaches can accommodate a wide variety of poses and have good localization ability (i.e. insensitive to misalignment), body part detection suffers from many false detections in the presence of severe background clutter or self-occlusion. Additionally, the use of 2D representation of poses (e.g. pictorial structures) [17], [19], [20], [21], [22], [23] and face and skin color [18], [19] restrict the range of applications.

Our approach does not explicitly fall into either approach, using both discriminative inference and optimization. Furthermore the overall framework is significantly different to either class of approaches. Most similar to our work is that of Andriluka et al. [24], where plausible configurations of body parts are found using a voting-based method. However, their work is limited to coarse pose estimation of walking people.

III. CANDIDATE GENERATION

The candidate generation is based on the use of a patch database which contains numerous model patches, i.e. patches extracted from silhouettes generated by a 3D human model using different pose parameters. We use these model patches as a training set of labeled examples, each of which is expressed by a set of target parameters $\theta$ and appearance $x$. Given an input image, we extract input patches, i.e. patches from the input image, and for each input patch, we compute the distance in the appearance space to every model patch. Then each input-model patch pair casts votes in the parameter space. Modes in the parameter space that are larger than a predefined threshold are selected as a set of candidates. We first describe the construction of the patch database and then explain patch retrieval, followed by a description of the voting scheme to obtain the set of candidates.

A. Constructing a Patch Database

We use a 3D human model consisting of truncated circular cones and a spheroid, as shown in Fig. 2(a), as the basis for the patch database. Each limb (an arm or leg) has four parameters, three joint angles for each shoulder or thigh, and one joint angle for each elbow or knee. We denote a set of limbs’ parameters as $y = (y_1, y_2, y_3, y_4)$, where $y_1, y_2, y_3$ and $y_4$ represent a set of pose parameters of the left arm, right arm, left leg and right leg, respectively. To accommodate different body types, a physique parameter, $f$, controls the radius of all truncated circular cones and spheroid. In total, our human model has 17 DOF. We limit the range of each parameter in order to prevent physically unrealistic poses.

To construct a model patch database, we render $N_a$ different silhouette images according to sets of randomly sampled pose parameters. For each rendered image, we randomly sample $N_b$ points from the contour of the silhouette to define the locations of the patches. In our approach, each patch is a constant sized square region and its appearance $x$ is a binary mask of the rendered silhouette in the patch as shown in Fig. 2(a). Thus $x$ is a bit string whose length is equal to the area of the patch. For each extracted patch, the vector $v$ indicates the position of the model’s reference point relative to the patch’s center. Additionally, the area of each limb $a = (a_1, a_2, a_3, a_4)$ within the patch are calculated.

To summarize, the patch database contains $N_{db} = N_a \times N_b$ patches, each of which is represented as $(x_n, \theta_n, a_n)$, where $\theta_n = (v_n, f_n, y_n)$ and $n \in \{1 \ldots N_{db}\}$.
B. Patch Retrieval

Given an input image, we first apply mean-shift segmentation and then randomly sample $N_{\text{input}}$ points from the segment borders to determine the input patches location $p_n$, where $n \in \{1 \ldots N_{\text{input}}\}$. $N_{\text{input}}$ is defined to be proportional to the total length of the segment borders in the input image. The input patch is the same size and shape as a model patch. Unlike a model patch which is a bit string indicating the silhouette mask, each input patch is represented as a collection of segment masks inside the patch. Suppose that input patch $n$ contains $S_n$ segments. The patch’s appearance is denoted as $X = \{x_1, \ldots, x_i, \ldots, x_{S_n}\}$, where each $x_i$ is a bit string representing $i^{\text{th}}$ segment mask. (Fig. 2(b))

The distance function between an input and model patch in appearance space captures how well the silhouette mask $x$ matches segment pieces $X$, as shown in Fig. 2(b), giving

$$d(X, x) = \sum_{i=1}^{S_n} \min(\text{bitcount}(x_i \land x), \text{bitcount}(x_i \land \bar{x})),$$

(1)

where \text{bitcount} is a function counting the number of 1 bits in a given bit string. As the most of the computation is done by bit operations, the computation is fairly fast.

For each input patch, distance $d$ is computed to every model patch and then converted to weight $w$ using $w = \exp(-K \cdot d)$, where $K$ is a predefined constant. As we shall explain in Sec. III-C, these input-model patch pairs form a set of weighted votes for the target parameter.

C. Voting Procedure and Formulation

We first give a new probabilistic formulation of the voting scheme for object detection, followed by its extension to joint human detection and pose estimation. We then present an efficient strategy to generate candidates in location-pose space using non-parametric mode seeking and propagation. Finally, some implementation details are presented.

1) Probabilistic Formulation of Voting-based Detection:

In voting-based object detection, a set of local patches, each of which consists of its own appearance and position relative to the object center, are collected from the training dataset to form a codebook for the object. During recognition, local patches are extracted from an input image and matched to the codebook, then each of the activated codebook entries casts votes for possible positions of the object center. Then the object is detected by finding a region which acquires more than a certain number of votes.

In order to interpret the voting-based detection in a probabilistic framework, most of the previous work [9], [25] represents the location of the object as a single variable $l$ and model a probability distribution $P(l)$ over the image. However, as also pointed out in [26], this assumes that there is only one instance of the object, hence creating a normalization problem if multiple objects exist. What we really want to know is whether or not there is a object at a given location $l$ in the image. This notion is adopted in most of the object detectors based on sliding-window search, which decides whether a given window contains the object of interest.

We formalize voting based object detection as follows: given a location $l$, the probability of $l$ being the reference point of the object is

$$P(r = 1|l) = \frac{1}{Z_1} \sum_{i=1}^{N_{\text{input}}} \sum_{j=1}^{N_{\text{db}}} w_{ij} G(l, p_i + v_j),$$

(2)

where $r \in \{0, 1\}$ indicates whether an object whose reference point is $l$ exists, $Z_1$ is a normalization constant to ensure that $\forall l; p(r = 1|l) \leq 1$, $w_{ij}$ is a weight calculated from the distance between $i^{\text{th}}$ input patch and $j^{\text{th}}$ model patch, $G$ is a Gaussian kernel with a predefined constant variance and $p_i$ is a position of the $i^{\text{th}}$ input patch. Note that codebook representation is a special case of database representation, thus codebook representation can easily be substituted for database representation by clustering model patches and allowing cluster centers to vote for multiple locations of the object.

In an object detector based on sliding-window search, image features of every possible window must be computed. However in the voting scheme, once feature computation and voting are done, we can compute $P(r = 1|l)$ by only accumulating votes. Furthermore unlike a typical sliding-window object detector, the efficient mean-shift mode search

Fig. 2. Patch construction and retrieval: (a) 3D human body model; (b) distance between input patch and model patch; and (c) distance between two model patches.
algorithm can be used to find likely locations of the objects, avoiding the need to examine all possible locations in the input image.

2) Extension to Pose Estimation: A straightforward extension to determine both human locations and poses would conduct voting and mean-shift mode search in joint space consisting of \( l, f \) and \( y \) at once. This fails due to the high dimensionality of the space. By factorizing the joint probability distribution into several conditional probability distributions, the problem is reduced to several manageable size problems, where we can perform mean-shift mode search.

First, the joint probability is factorized into a product of two probabilities

\[
P(y, f, r=1|l) = P(y|f, r=1, l)P(f, r=1|l) .
\]

As the dimensionality of \( y \) is still too large (16DOF), assuming that the probability distribution of each limb’s pose is statistically independent gives

\[
P(y|f, r=1, l) = \prod_{p=1}^{4} P(y_p|f, r=1, l) .
\]

Finally, the joint probability that there exists a person with pose \( y \) and \( f \) given a location \( l \) is factorized as:

\[
P(y, f, r=1|l) = \frac{1}{Z_1} \sum_{i=1}^{N_{\text{input}}} \sum_{j=1}^{N_{\text{th}}} w_{ij}G(l \parallel cf), ((p_i + v_j) \parallel cf) ,
\]

where \( \parallel \) represents vector concatenation and \( c \) is a predefined constant to adjust physique to the scale of location space.

To compute each \( P(y_p|f, r=1, l) \), votes in \( y_p \) space from those which have voted to the point whose distance to the given \( l \) and \( f \) is smaller than a predefined distance \( R \) are accumulated. Formally, \( P(y_p|f, r=1, l) \) is defined as

\[
P(y_p|f, r=1, l) = \frac{1}{Z_2} \sum_{i=1}^{N_{\text{input}}} \sum_{j=1}^{N_{\text{th}}} w_{ij}K(l \parallel cf, ((p_i + v_j) \parallel cf) ) G(y_p, y_{p,j}) ,
\]

where \( y_{p,j} \) represents the vote from \( j \)th model patch into \( y_p \) space, \( K \) is the triangular kernel \( K(z_1, z_2) = (1 - |z_1 - z_2|/R) \) for \( |z_1 - z_2| \leq R \) and 0 otherwise, and \( Z_2 \) is a normalization constant.

3) Candidate Generation: Based on (6), we first find modes in \( l \parallel f \) space using mean-shift. From each probability distribution \( P(y_p|f, r=1, l) \) conditioned on the found modes, we find modes again using mean-shift. Location-pose candidates are obtained as all possible combinations of modes in \( l \parallel f \) space and corresponding modes in \( P(y_p|f, r=1, l) \).

4) Implementation Details: For further reduction of computation time without an unacceptable drop in accuracy, first, votes with a small weight \( w \) are ignored, which is equivalent to retrieving model patches whose distance to the input patch \( d \) is smaller than a certain threshold. Also the number of votes per each input patch is limited to within a predefined threshold \( V \). (e.g. \( V = 10 \)). These procedures significantly reduce the computation time of the mean-shift mode search at all stages.

Second, a threshold is set to discard small modes in both location and limb’s pose spaces. If the number of obtained modes in each limb’s pose space is larger than \( T \), the highest \( T \) modes are selected. This restricts the number of candidates per mode in location-physique space to less than \( T^4 \).

Finally, when voting in \( y_p \), if the area of limb, \( a_p \), of each model patch is lower than a threshold, we restrict this patch from voting for \( y_p \) as a model patch which contains little information about \( j \)th limb cannot provide meaningful information to estimate \( y_p \). Thus, we can exclude unreliable votes, allowing both higher computation efficiency and accuracy.

IV. Top-down Verification of Candidates

In this step, each candidate is thoroughly verified using a top-down model-based method in conjunction with a learned scoring function. Verification starts by projecting the 3D model of each candidate onto the input image and computing several types of scores capturing different features of the candidate. Then a trained SVM regression aggregates these scores into a single score.

A. Silhouette Matching Score

Similar to the distance function between an input patch and a model patch (1), the silhouette matching score captures how well a silhouette fits to segment pieces. However, the entire image is treated as one patch in computing the silhouette matching score. We represent candidate’s silhouette by a binary string \( x \) whose length is equal to the area of the image. Supposing there are \( S \) segments in the segmented input image, we represent a collection of each segment mask as \( X = \{x_1 \ldots x_S\} \). As with (1), we can calculate

\[
d_s(X, x) = \sum_{i=1}^{S} \min \left( \text{bitcount}(x_i \land x), \text{bitcount}(x_i \land \bar{x}) \right) .
\]

The silhouette matching score is then

\[
sil = \exp(-d_s) .
\]

B. Color Dissimilarity Score

Intuitively the appearance of the foreground tends to be different from that of the background. In order to capture this property, we compute a normalized color histogram (measured in Lab space) both inside as well as outside the silhouette and compute the Bhattacharyya coefficient \( BC_d \) between these two histograms. The \( BC_d \) is directly used as the color dissimilarity score.
C. Limbs Symmetry Score

The appearance of corresponding limbs (e.g., left and right arms) tends to be symmetric, allowing calculation of the color similarity between regions of both arms, $BC_{\text{arm}}$, and both legs, $BC_{\text{leg}}$, using a normalized color histogram and the Bhattacharyya coefficient. We directly use $BC_{\text{arm}}$ and $BC_{\text{leg}}$ as the limbs symmetry scores.

D. Final Score Computation

Although these subscores can offer clues to decide if a candidate should be accepted as a human, we need to decide how to aggregate these scores into a single score which we use to make a final decision. If we have a training dataset comprising a set of subscores (input values) with corresponding final scores (target values), we can obtain the decision rule using supervised learning. In our case, an ideal target value is the similarity between a candidate location-pose pair and ground truth location-pose pair. However due to the difficulty in obtaining ground-truth data of real human poses, we employ an approximation using the ratio of overlap between a ground truth silhouette and a silhouette generated from a candidate as a target value. Specifically, the ratio is computed as the area of intersection of the two silhouettes divided by that of union of the two silhouettes.

In order to create a training dataset, we first collect images of people with a wide range of poses, clothing styles and backgrounds. We then manually extract the ground truth silhouette of human from each image and store them along with the original image. Subsequently, we run our algorithm on the original images to obtain a set of candidates with corresponding subscores and target values. Finally, we train $\varepsilon$-support vector regression ($\varepsilon$-SVR) [27] with RBF kernel, owing to its better generalization capability in a nonlinear space.

V. Experiments

First we conduct experiments to evaluate the detection and pose estimation performance of our algorithm, where the algorithm detects multiple people from each given image and estimate each subject’s pose. The detection performance is evaluated quantitatively while we show some qualitative results of the pose estimation. We then focus on pose estimation task where the algorithm is given images, each of which is guaranteed to contain only one subject, and outputs the candidate with the highest final score. Throughout the experiments, the number of model’s silhouette $N_a$ is set to 3,000 and the number of model patches extracted from each model silhouette $N_b$ is set to 100, resulting in $N_{db} = 300,000$. Note that in typical example-based pose estimation methods such as [6] and [4], each example in a database corresponds to an entire pose, thus a huge number of pose examples is required to deal with a large variation of possible poses (e.g., 150,000 pose example are needed only for upper body poses in [6]). Whereas, our method requires a small number of poses ($N_a = 3,000$), yet achieves good results as we will show.

A. Detection and Pose Estimation

We select 67 images containing only people facing toward the camera from 288 positive images for the testing of the INRIA Person Dataset. In order to evaluate detection performance, we manually annotate each person in the images by a bounding box. A detected bounding box is considered correct if it overlaps with the ground truth bounding box more than a predefined threshold. Overlap is measured by “intersection over union”. As for the threshold, we first use 50% criterion which is widely employed for object detection literature. We also use a more strict 65% criterion to confirm the better localization ability of our method over the window search based object detector. More strict thresholds than 65% is not tested as we find the use of bounding boxes becomes less reliable.

We run our algorithm at different scales with a stride value of 1.05 and collect every candidate whose final score is higher than a predefined threshold. To suppress multiple detections overlapping each other, we employ a greedy approach where the candidate with the highest score is accepted and the silhouette of the candidate is rendered on the input image. Then the candidate with the second highest score is accepted only if the area of overlap between the already rendered silhouette and the silhouette of the candidate is below a predefined threshold. This procedure is repeated until all candidates are tested. Finally the algorithm outputs bounding boxes according to the locations of the reference points as well as the scale of the accepted candidates.

We plot Precision-Recall (PR) curves\(^1\) of our detection along with that of the publicly available HOG pedestrian detector [10] for both 50% and 65% criteria (Fig. 3). We use $T = 2$ for our method. (See Implementation Details in Sec. III-C) Consequently, HOG outperforms our method on 50% criteria while both produce comparable results on 65% criteria. False negatives of our method are mostly due to fault segmentation which incorrectly merge foreground and background regions into one segment. False positives mostly appeared in densely segmented areas such as trees and structures resembling human shapes. Incorporating object detector with our method might improve the overall performance.

Needless to say, the biggest benefit of our method is that the pose of the detected person can be obtained simultaneously. In Fig. 4, we show some sample results of our method at a recall of 80%. Despite the cluttered backgrounds and a wide range of poses, our method successfully reconstruct the poses.

B. Pose Estimation Performance

In order to evaluate the pose estimation performance of our method quantitatively, we use synthetic human foreground images of several figures with known joint positions using a human model rendering package, POSER. We create a test dataset ‘noisy’ by placing each of these foregrounds on a random location in front of a background drawn from a set

\(^1\)The annotation and evaluation tool provided by P. Dollar is used. http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/
Fig. 4. Sample detection and pose estimation results obtained on the subset of the INRIA Person Dataset at a recall of 80%. The lower right result includes some false detections. Notice that the physique of the subjects are also estimated (see also Fig. 6).

Fig. 3. Precision-Recall curves of detection results: (a) results from 50% criterion, and (b) results from 65% criterion. HOG’s performance deteriorates due to stricter criterion for matching while our method maintains its performance.

We also run a pose estimation software obtained from Eichner and Ferrari [21] on both datasets. Their algorithm uses a 2d pictorial structure framework and achieves state-of-the-art performance on the “Buffy” dataset [23]. As their current implementation is restricted in upper body pose estimation, we only evaluate the performance for the upper body joints. As shown in Table II, our algorithm outperforms their method although a strict comparison is difficult as two methods are trained on different training data. Note that our method can also estimate 3d full body pose of the subjects in the form of joint angles, which is usually preferable to the positions of the joints.

Finally we show some sample results run on real images from [17] in Fig. 6. We show the original images, results of the proposed method and results of [21].

As shown in Table I, top-down verification largely reduces the error on the ‘noisy’ dataset while its effect on the ‘clean’ dataset is negligible, demonstrating the efficacy of the top-down verification on images with background clutter. The result on the ‘noisy’ is worse than that of ‘clean’ as we expect although the difference in error is not very large despite the huge difference in the amount of noise in background. The big difference between the 2d and 3d RMS errors is due to the inherent ambiguity of predicting depth information from only 2d information (i.e. images). Exploiting prior in pose space is likely to reduce the ambiguity. Sample results are shown in Fig. 5.

<table>
<thead>
<tr>
<th></th>
<th>No top-down verification</th>
<th>$T = 1$</th>
<th>$T = 2$</th>
<th>$T = 3$</th>
<th>$T = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean 2d</td>
<td>8.5</td>
<td>8.4</td>
<td>8.4</td>
<td>8.4</td>
<td>8.3</td>
</tr>
<tr>
<td>Clean 3d</td>
<td>32.4</td>
<td>32.4</td>
<td>32.1</td>
<td>32.3</td>
<td>31.8</td>
</tr>
<tr>
<td>Run-time</td>
<td>1.3</td>
<td>1.4</td>
<td>2.0</td>
<td>13.8</td>
<td>21.9</td>
</tr>
<tr>
<td>Noisy 2d</td>
<td>14.5</td>
<td>14.1</td>
<td>12.8</td>
<td>12.0</td>
<td>11.8</td>
</tr>
<tr>
<td>Noisy 3d</td>
<td>36.4</td>
<td>36.2</td>
<td>35.2</td>
<td>34.4</td>
<td>34.4</td>
</tr>
<tr>
<td>Run-time</td>
<td>5.9</td>
<td>6.3</td>
<td>10.8</td>
<td>35.2</td>
<td>113</td>
</tr>
</tbody>
</table>
TABLE II

<table>
<thead>
<tr>
<th></th>
<th>Our method</th>
<th>Eichner and Ferrari [21]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean 2d</td>
<td>9.7</td>
<td>10.4</td>
</tr>
<tr>
<td>Noisy 2d</td>
<td>13.6</td>
<td>17.4</td>
</tr>
</tbody>
</table>

In this paper we demonstrated that our voting based pose estimation method is capable of seamlessly localizing people in images and estimating their poses. Our method has advantages in accommodating a wide variety of poses as it can construct previously unseen poses by assembling local patches extracted from different locations of various poses. This approach is similar in spirit to the class of approaches which first detect body parts candidates and then find the most plausible configuration of the parts. The main difference is that our local patches cast multiple votes with varying weights for possible locations of people as well as their poses, producing a complex probability distribution of the pose in a non-parametric manner. The mode seeking in the high dimensional probability distribution is efficiently done using a multi-stage mean-shift procedure, producing location-pose candidates. Owing to the robust description of the patches as well as the distance measure, our method works well on images with severe background clutter or self-occlusion. The holistic top-down verification of the candidates is shown to significantly improve the performance.

VI. CONCLUSION

In this paper we demonstrated that our voting based pose estimation method is capable of seamlessly localizing people in images and estimating their poses. Our method has advantages in accommodating a wide variety of poses as it can construct previously unseen poses by assembling local patches extracted from different locations of various poses. This approach is similar in spirit to the class of approaches which first detect body parts candidates and then find the most plausible configuration of the parts. The main difference is that our local patches cast multiple votes with varying weights for possible locations of people as well as their poses, producing a complex probability distribution of the pose in a non-parametric manner. The mode seeking in the high dimensional probability distribution is efficiently done using a multi-stage mean-shift procedure, producing location-pose candidates. Owing to the robust description of the patches as well as the distance measure, our method works well on images with severe background clutter or self-occlusion. The holistic top-down verification of the candidates is shown to significantly improve the performance.

REFERENCES