IEICE TRANS. INF. & SYST., VOL.E97–D, NO.10 OCTOBER 2014

PAPER

Improving Hough Based Pedestrian Detection Accuracy by Using Segmentation and Pose Subspaces

Jarich VANSTEENBERGE[†a)], ***Nonmember***, **Masayuki MUKUNOKI**[††], ***Member***, ***and*** **Michihiko MINOH**[††], ***Fellow***

SUMMARY The Hough voting framework is a popular approach to parts based pedestrian detection. It works by allowing image features to vote for the positions and scales of pedestrians within a test image. Each vote is cast independently from other votes, which allows for strong occlusion robustness. However this approach can produce false pedestrian detections by accumulating votes inconsistent with each other, especially in cluttered scenes such as typical street scenes. This work aims to reduce the sensibility to clutter in the Hough voting framework. Our idea is to use object segmentation and object pose parameters to enforce votes' consistency both at training and testing time. Specifically, we use segmentation and pose parameters to guide the learning of a pedestrian model able to cast mutually consistent votes. At test time, each candidate detection's support votes are looked upon from a segmentation and pose viewpoints to measure their level of agreement. We show that this measure provides an efficient way to discriminate between true and false detections. We tested our method on four challenging pedestrian datasets. Our method shows clear improvements over the original Hough based detectors and performs on par with recent enhanced Hough based detectors. In addition, our method can perform segmentation and pose estimation as byproducts of the detection process.

key words: *Hough based detections, pedestrian segmentation, pose estimation, Random Forest, kPCA*

1. Introduction

Since the early days of computer vision, researches related to object recognition have received a large amount of interest. Whether it be faces, letters, humans or cars, the ability to identify an object and to isolate it from its environment is the building block of any computer vision based system.

Roughly speaking, object recognition can be divided into three main areas, namely, object detection, object segmentation and object's pose estimation. Arguably, object detection has received the most attention among the three research directions. Since the mid 90s pedestrian detection is considered a hot topic, which is still actively explored to this day. From the mid 2000s pedestrian detection datasets shifted from controlled environments to increasingly realistic street scenes where appearance changes and occlusions are omnipresent. Consequently, models built on combination of parts have gathered a lot of interest.

A popular approach to part based pedestrian detection is the pictorial structure (PS) [1] in which a limited set of parts and their relative spatial configurations are learned from training images. Detection is formulated as an optimization problem where the goal is to minimize the total cost of placing parts at given locations and the cost of violating the prior spatial configuration. PS models using visually discriminant parts [2] led to state of the art human detection performances on various datasets. PS models using kinematic parts such as head, arms, legs and torso [3] produced state of the art results for human pose estimations. Despite its current popularity, the PS approach suffers from two weaknesses. First, the inference complexity limits the number of parts that can be handled. Second, the detection being constrained by a global structure of a few parts, it is very sensitive to mild occlusions.

A second popular approach to part based pedestrian detection is the Generalized Hough Transform [5]. One of the most influential methods based on the Hough transform is the implicit shape model (ISM) [4] of Leibe et al. which models the distribution of object's parts over a star shaped spatial structure. The detection process relies on the Hough voting framework. Parts extracted from a test image are allowed to vote for the positions and scales of pedestrians within the image. All the votes are gathered into an accumulator before the algorithm looks for locations which have gathered enough votes. Each of these locations constitutes a candidate's detection center position. Unlike the PS, each voting part in the Hough based detectors are considered independent from each other. This property allows parts to vote independently and thus, a potentially unlimited number

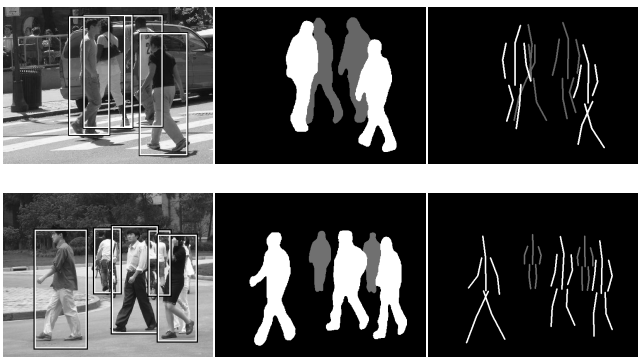


Fig. 1 Example results from our pedestrian detector. Detection results *(left)* with the produced segmentation masks *(middle)* and estimated poses *(right)*.

Manuscript received March 24, 2014.

[†]The author is with the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.

[††]The authors are with Academic Center for Computing and Media Studies, Kyoto University, Kyoto-shi, 606–8501 Japan.

a) E-mail: vansteenberge@mm.media.kyoto-u.ac.jp

DOI: 10.1587/transinf.2014EDP7092

of them could participate in the detection process. Moreover, the independence assumption prevents occluded parts to have a negative impact on the detection process. As a result, Hough based detectors are extremely robust to occlusions, and offer fast inference possibilities. However the assumption of independence is also responsible for the Hough based detectors' weaknesses. Indeed, the naïve accumulation of votes allows for combination of potentially incompatible parts to vote for the same center position, triggering a false detection. This is especially true on cluttered background, which typically matches a lot of parts subsequently casting a large amount of inconsistent votes. Hallucinated detections due to inconsistent votes are a recurrent problem with Hough based detectors. In recent years, a series of works have aimed to alleviate this problem.

We base our method on the Generalized Hough Transform framework. Our idea is to use segmentation and pose subspaces to enforce votes' consistency both at training and testing time.

2. Related Work

A simple way to enforce the consistency of votes is to condition the training of different models based on a global variable. More specifically, the idea is to partition the training data according to the training object's properties. A different model is trained on each group of data which insures that all votes coming from it will be consistent in regard to the global property. Dantone et al. [6] divide the training set based on faces' orientations. At test time, the detected face's rough orientation is estimated before the closest model is selected and allowed to vote for fiducial points locations. Similarly, Sun et al. [7] vote for human pose estimation by conditioning on the torso orientation and the person's height. These approaches have shown improvements over baseline methods, but require to first estimate the state of the object's property before the corresponding model is allowed to cast votes. Razavi et al. [8] generalized this approach to multiple conditioning variables.

Instead of enforcing consistency by training different models on subsets of data, other works propose to extend the Hough space with the conditioning variable. As a result, each cluster of votes in the Hough space will be consistent in regard to both the detection parameters and the conditioning variable. Among others, Hough space can be extended with depth information [9], viewpoint orientation [10] and object's bounding box ratio [11].

From the literature, it is still unclear which strategy performs the best at enforcing votes' consistency. Learning different models on subsets of data allows to use simpler models and to tune each of them to the particularities of each subset. However, each model doesn't make full use of the available training data. On a contrary, learning a single model to vote into extended subspaces makes better use of the training data, but the large increase in time and memory consumption due to the higher Hough space dimensions makes such approaches less practical.

A common feature of all these works is that they are trying to improve the votes' consistency prior to the detection. A different approach consists in verifying the consistency posterior to the detection. This idea was introduced in the original work of Leibe et al. [4] where objects' estimated segmentation masks are used to verify detections' hypothesis. Simply put, a segmentation patch is attached to each visual words and used at test time to infer the detected object's rough segmentation mask. The region defined by this mask is used to discard votes coming from regions outside of the estimated foreground. This means that votes which are inconsistent with the global explanation provided by other votes will be discarded. The final masks are used in a Maximum Description Length verification scheme and have shown significant improvements in the detection accuracy.

More recently, Wohlhart et al. [12] have explored the potential of hypothesis verification. They use activation vectors which describe which votes have been activated or not and their respective weights in the current detection. A set of true and false detections' activation vectors is collected and used to train a non-linear SVM. At test time, the candidate detection's activation vectors are classified by the SVM to determine if there are true or false detections. Their methods have shown improvements over the baseline detector, however they mentioned the need for several rounds of bootstrapping on the validation set to obtain good results. Thus, it is unsure if the improvements are due to the SVM having learned to recognize failure pattern of their detector or if the SVM is now capable of measuring the level of consistency between the votes.

In this work, we propose to use segmentation and pose parameters to enforce votes' consistency, both prior and posterior to the detection process. Segmentation and pose estimation are so intrinsically related to object detection that they can act as road guards for the detection. We introduce segmentation and pose parameters to learn a pedestrian model which can cast votes consistent in terms of detection, segmentation and pose. Furthermore, we use segmentation and pose subspaces as bases to measure votes' consistency for hypothesis verification. In other words, we measure the detections' quality from both segmentation and pose standpoints. Our method is summarized as follows.

We initially condition the training of multiple models based on the viewpoint. Then during the training process, the training samples are progressively grouped according segmentation and pose parameters. At test time, no preselection of the best model is made, we simply allow all models to votes for the object detection's parameters. The candidate detections' segmentation and pose votes' densities are then measured and used to perform hypothesis verification. We show that these simple measurements of votes' consistency in pose and segmentation subspaces can help discriminate between true and false detections.

Our contributions are as follows:

- In addition to the viewpoint, we propose to use seg-

mentation and pose parameters to condition a Hough Forest.
- We propose a hypothesis verification step based on pose and segmentation subspaces which can provide simple yet efficient feedbacks to improve Hough based pedestrian detection.
- We propose an efficient way to produce full body segmentation masks and full body pose estimations as a byproduct of the detection process.

3. Method

3.1 Conditional Hough Forest for Pedestrian Detection

Hough Based detectors estimate a target object's detection parameters by collecting a set of votes into an accumulator. Local features $f_k \in I$ are extracted from a test image at location l_k and matched against a learned representation $\{L_m\}_{m=1}^{M}$ of the object's class. Each matched element L_j will casts a vote $v(\mathbf{h}|f_k, l_k, L_j)$ for the object hypothesis $\mathbf{h} = (h_x, h_y, h_s)$ where (h_x, h_y) is the hypothesis location within the 2D image and h_s is its scale. The score for a given hypothesis is the sum over all votes:

$$S(\mathbf{h}) = \sum_k \sum_j v(\mathbf{h}|f_k, l_k, L_j). \tag{1}$$

Hypotheses whose scores are over a given threshold are estimated object's locations and scales within the test image.

To implement a Hough based detector, the target object's class representation and its associated votes needs to be learned from training images. We base our pedestrian detector on the Hough Forest [11] of Gall and Lempitsky which consists of random trees optimized to vote for object detection. We enhance our forest by using viewpoints, segmentation and pose parameters as conditioning parameters during the training. The general procedure to train and test our detector is described bellow.

At training time, local patches are extracted from random locations within negative images and from the object's bounding box for positive images. Each training patch $P_i = (\mathbf{A}_i, \mathbf{s}_i, \mathbf{p}_i, c_i, \mathbf{d}_i)$ consists of, a local appearance features vector $\mathbf{A}_i$ (detailed in Sect. 4.1), full body segmentation parameters $\mathbf{s}_i$ and full body pose parameters $\mathbf{p}_i$ (described in Sect. 3.3), a class label $c_i \in \{0, 1\}$ ($c_i = 1$ if the patch is from a positive training image) and an offset to the pedestrian center $\mathbf{d}_i$. For negative patches, the offset vectors as well as the pose and segmentation parameters are left undefined. Our feature vectors $\mathbf{A}_i$ are extracted at a fixed resolution but also at half the current resolution. As a result $\mathbf{A}_i$ not only captures the local patch appearance, but also the direct surrounding area which provided contextual information.

Each tree is grown by recursive splitting of the training patches. Starting from the root node, the incoming patches $Z = \{P_i\}$ are split into two subsets Z_l and Z_r according to a binary test $t_q(\mathbf{A}_i) \rightarrow \{0, 1\}$ which simply compares the values of two random positions within the feature vectors. At each node 1000 random binary tests $\{t_q\}$ are created and the corresponding splits qualities are estimated according to one of the following uncertainty measurements:

$$U_1(Z_t) = -\sum_{i=1}^{|Z_t|} p(c_i) log(p(c_i)), \tag{2}$$

$$U_2(Z_t) = \sum_{i:c_i=1} \left\|\mathbf{d}_i - \overline{\mathbf{d}}\right\|^2, \tag{3}$$

$$U_3(Z_t) = \sum_{i:c_i=1} \left\|\mathbf{s}_i - \overline{\mathbf{s}}\right\|^2, \tag{4}$$

$$U_4(Z_t) = \sum_{i:c_i=1} \left\|\mathbf{p}_i - \overline{\mathbf{p}}\right\|^2, \tag{5}$$

where, Z_t is a subset of patches leaving a splitting node, $\overline{\mathbf{d}}$, $\overline{\mathbf{s}}$ and $\overline{\mathbf{p}}$ are the mean offset vector, mean segmentation and mean pose in Z_t, $p(c_i)$ is the probability of having $c_i = 1$ withing Z_t. By minimizing such uncertainty measurements, we maximize the consistency between patches in Z_t. The first measurement U_1 tends to produce subsets of patches sharing identical class labels. The second uncertainty U_2 regroups patches having similar locations relative to the pedestrian center. The third U_3 and last measurement U_4 tend to gather patches extracted from images with pedestrian sharing similar segmentations and similar poses. Once the current node's uncertainty measurement U_s has been randomly selected from U_1 to U_4, we look for the split test which maximize the subsets' patches consistencies according to:

$$\min_{t_q} \left(\frac{|Z_l|}{|Z_l| + |Z_r|} U_s(|Z_l|) + \frac{|Z_r|}{|Z_l| + |Z_r|} U_s(|Z_r|) \right), \tag{6}$$

The two subsets Z_l and Z_r are then passed on to child nodes where recursive splitting is performed until the number of patches within an incoming set is smaller than 10. In such case, a leaf node is created and associated patches' information is collected, that is, the ratio ω between the number of positive patches over the total number of patches, the offset vectors $\{\mathbf{d}_i\}$ and the set of segmentation and pose parameters $\{\mathbf{s}_i\}$ and $\{\mathbf{p}_i\}$. The training finishes once every tree has been grown up to its leaves. The resulting HF consists of a series of binary tests from roots to leaves, which models the local appearance of pedestrians sharing similar poses and segmentations.

At test time, image's local patches $f_k \in I$ are densely sampled from locations l_k in a test image. Each local patch is then matched against the forest. Starting from the root, the local patch appearance is subjected to binary tests and passed down the trees up until it reaches leaves nodes. A forest of T trees will provide T matching leaves $\{L_t\}_t^T$ for a single test patch f_k. Each matching leaf will then casts votes according to the patch information it collected at training time $(\omega, \{\mathbf{s}_i, \mathbf{p}_i, \mathbf{d}_i\}_i^I)$. Specifically a leaf casts weighted votes in the accumulator at locations $(h_x, h_y)_i = l_k - \mathbf{d}_i$ with a weight $\beta = \omega/|I|$. Votes are accumulated for every matching leaf and every test patch extracted from the test image.

Once all the votes have been collected we search for candidate detections as peaks in the accumulator, that is, the set of locations $\{(h_x, h_y)_c\} = \{(h_x, h_y)|S(h_x, h_y) > \tau\}$, where $S(h_x, h_y)$ is the score for a detection at location (h_x, h_y) computed as the sum of votes' weights at this location, and τ the detection threshold. These candidate detections are then submitted to hypothesis verification described in Sect. 3.4

Our forest differs from [11] in 3 ways:

- We employ segmentation and pose parameters to guide the object's parts learning process. This conditioning strengthens the consistency of votes coming from a single leaf which helps to improve the detection accuracy.
- We use a contextual representation. Our forest makes natural use of local and contextual information which reduce the number of matches on clutter.
- Our detector can vote for the pedestrian segmentation and pose associated with the current detection.

The following sections present how we obtain full body segmentations and full body pose parameters from annotated data.

3.2 Full Body Segmentation and Pose Parameters

Our goal is to describe each training set sample's full body segmentation mask $\mathbf{x}_i^s$ and full body pose $\mathbf{x}_i^p$ as parameter vectors $\mathbf{s}_i$ and $\mathbf{p}_i$ respectively. To do so, we chose to learn two generative models able to synthesize full body segmentations and full body poses. These models' respective parameters $\mathbf{s}_i$ and $\mathbf{p}_i$ will act as low dimensions descriptors for the ground truth training set annotations.

In order to capture the underlying structures of segmentation and pose generation process, we employ a nonlinear extension of the PCA algorithm called kernel PCA [13]. Given the input data $X = \{\mathbf{x}_1 \dots \mathbf{x}_N | \mathbf{x}_i \in \mathbb{R}^D, i = 1 \dots N\}$, the kernel PCA aims to find an alternative set of orthogonal principal axes $\mathbf{V}$ with which to describe the data. The idea is to first map the original data into a high dimensional space $\mathcal{F}$ via a nonlinear function $\Phi : \mathbb{R}^D \to \mathcal{F}, \mathbf{x} \mapsto \Phi(\mathbf{x})$, then to perform traditional PCA on the projected data. That is, solving the eigenproblem $\boldsymbol{\lambda}\mathbf{V} = \overline{C}\mathbf{V}$ with eigenvalues $\lambda \geq 0$ and eigenvectors $V \in \mathcal{F}\backslash\{0\}$. Here $\overline{C} = (1/N)\sum_{i=1}^{N}(\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T)$ is the covariance matrix of the data centered in feature space $\mathcal{F}$. All solutions in $\mathbf{V}$ must lie in the span of $\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_N)$ and thus there must be coefficients α_i^k such as:

$$V^k = \sum_{i=1}^{N} \alpha_i^k \Phi(\mathbf{x}_i). \tag{7}$$

Introducing the Gram matrix G with $G_{ij} = \left\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \right\rangle$ [13] have shown that the eigenproblem becomes $N\boldsymbol{\lambda}\boldsymbol{\alpha} = G\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the column vector of unknown coefficient α_i that will form the new basis. The projection of a point $\mathbf{x}$ from the original space onto an eigenvector V^k in $\mathcal{F}$ is:

$$\left\langle V^k, \Phi(\mathbf{x}) \right\rangle = \sum_{i=1}^{N} \alpha_i^k \left\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \right\rangle. \tag{8}$$

The set of K eigenvectors V^k having the largest variances are selected to form the new orthogonal basis to represent the input data. This subspace aims to retain most of the interesting information from the data while hopefully getting rid of the noise.

Equation (8) requires to work with high dimensional vectors which is computationally expensive and might be intractable. By using the kernel trick, it is possible to evaluate the dot product between the mapped high dimensional vectors without actually computing the projections. Any positive semidefinite kernel $k(\mathbf{x}, \mathbf{y})$ can be used to replace dot products in (8) as $k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$ which makes the kPCA algorithm practical.

3.3 Segmentation and Pose Subspace

We now can learn both subspaces by using kPCA and our dataset segmentation annotations and pose annotations. We chose to apply kPCA with a Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/2\sigma^2)$. Hence, each model comes with two parameters to be determined, the optimal dimensions for the subspace K and the kernel bandwidth σ. Because segmentation and pose parameters will be used to grow our forest and will be stored within its leaves, it is important to limit their dimensions to reduce computational costs and the forest's memory footprint. Therefore, we use an iterative approach where at each step, the number of dimensions is increased and the best performing kernel bandwidth is determined.

Specifically, we start with a single dimension $K = 1$, and look for the optimal kernel bandwidth. For each couple (K, σ), kPCA is applied to learn the segmentation and pose subspaces. The training samples are then projected and back-projected to and from these subspaces. The Original samples and their back-projected versions are compared in order to asses the ability of the subspaces to preserve segmentation and pose information. The best performing couple $(\hat{K}, \hat{\sigma})$ is stored before the number of dimensions in increased by one. This process is repeated until the best couple provides performance satisfying stopping criteria. We set up these criteria such that the mean precision and mean recall for the dataset's back-projected segmentations are over 90%, and the mean reconstruction error is under 15 pixels for the reconstructed poses. The projections of annotations to the learned subspaces are performed as:

$$\begin{aligned} \mathbf{s}_i &= \left[\left\langle V_s^1, \Phi(\mathbf{x}_i^s) \right\rangle, \dots, \left\langle V_s^K, \Phi(\mathbf{x}_i^s) \right\rangle \right]^T, \\ \mathbf{p}_i &= \left[\left\langle V_p^1, \Phi(\mathbf{x}_i^p) \right\rangle, \dots, \left\langle V_p^K, \Phi(\mathbf{x}_i^p) \right\rangle \right]^T, \end{aligned} \tag{9}$$

where V_s^K, V_p^K are the segmentation and pose subspaces bases and $\mathbf{x}_i^s$, $\mathbf{x}_i^p$ are the i-th training sample's segmentation annotation vector and pose annotation vector respectively.

The back-projection from the subspaces to the original spaces is not a trivial operation as some points might not exist in the original spaces. This is known as the pre-image problem and is still an open issue. We use the method proposed in [15] which computes the pre-image approximation

Table 1 Distribution of training images over the 8 viewpoints.

Viewpoint °	0	45	90	135	180	225	270	315
Images	88	103	96	124	108	124	96	103

(a) (b) (c) (d) (e) (f)

Fig. 2 (a)(d) Example pose annotations. (b)(e) Original segmentation masks. (c)(f) Reconstructions of original masks using DCT coefficients.

z of a subspace point $\mathbf{o}$ as:

$$\mathbf{z}_{t+1} = \frac{\sum_{i=1}^{I} \gamma_i k(\mathbf{z}_t, \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{I} \gamma_i k(\mathbf{z}_t, \mathbf{x}_i)}, \tag{10}$$

with $\gamma_i = \sum_{k=1}^{K} o_k \alpha_i^k$. Here o_k the k'th element of the subspace point o. This is an iterative approach where the solution is gradually shifted toward training samples whose projections are close to the point we want to back-project. The reader is invited to read [15] for further details about the pre-image algorithm.

Annotation and training set: Our subspaces are assumed to be representative of pedestrian segmentation and pose spaces, so they can be used for all our experiments involving multiple views pedestrians. As a result, both subspaces are learned only once and re-used on the 4 pedestrian datasets we considered. We used the TUD multiview Pedestrian training data [14] as our training dataset. It consists of 2366 grayscale images of pedestrians divided into 8 subsets according to the viewpoints. We randomly selected 421 images from the dataset and used mirrored version to increase the number of training samples to 842. The samples distribution can be seen in Table 1.

We resized each training sample such that the pedestrian's height is approximately 160 pixels and we hand annotated it's segmentation mask and pose. Example pose annotations can be seen in Fig. 2. The pose annotation vectors $\mathbf{x}_i^p$ fed to the kPCA algorithm are vectorized 2D locations of annotated points $q_1, \ldots, q_{13}$ relative to the shoulders' 2D center point q_h, i.e. $\mathbf{x}^p = [(q_1 - q_h), \ldots, (q_{13} - q_h)]^T$ with $q_h = (q_2 - q_3)/2$.

We describe each ground truth segmentation mask with 20x20 discreet cosine transform (DCT) coefficients. Despite a loss in masks' precision, DCT coefficients are powerful descriptors able to retain a large amount of information over a limited number of parameters. The decrease in segmentation mask precision can be seen in Fig. 2. The resulting segmentation annotation vectors $\mathbf{x}_i^s$ have a dimension of 400. After applying kPCA in our iterative approach, the optimum segmentation subspace ended up with 16 dimensions and a kernel bandwidth of 30. The optimum pose subspace has 8 dimensions and a bandwidth of 65. Figure 3 shows the first 3 dimensions for both subspaces.

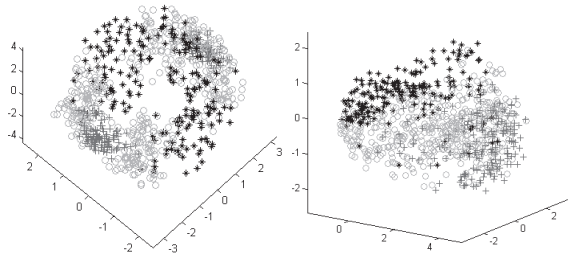


Fig. 3 Scatter plot of the annotated data on the first 3 dimensions of the pose subspace *(left)* and segmentation subspace *(right)*. Black stars are the side views. Gray crosses are the front and back views. Light gray circles are intermediate viewpoints.

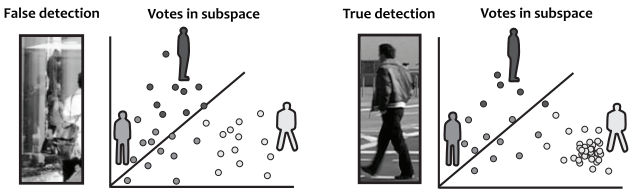


Fig. 4 Illustration of the consensus point. False detections due to clutter gather votes without a clear consensus. The presence of a real pedestrian triggers a large amount of votes agreeing with the real pedestrian configuration.

The next section describes how we use these subspaces to perform detection's hypothesis verifications.

3.4 Consensus Point

As mentioned in the introduction, aside from conditioning the training of the detectors on particular variables, Hough based detection accuracy can be improved by using hypothesis verification. In this work we aim to measure the amount of agreement within the votes supporting a candidate detection by looking at segmentation and pose parameters distributions. Intuition tells us that, if there are evidences supporting the presence of a real pedestrian at the current location and scale, then there must be a consensus among the votes regarding the segmentation and pose of the same pedestrian. This consensus translates as locations of high densities of votes within both segmentation and poses subspaces. We call these locations consensus points. If no clear consensus can be found between votes supporting the detections, then it is likely to be an accumulation of votes from parts unrelated to each other. This idea is illustrated in Fig. 4.

Our hypothesis verification step is performed as follows. For each candidate detection $\mathbf{h} = (h_x, h_y, h_s)$ we recover the list of its supporting votes $\{v(\mathbf{h}|f_k, l_k, L_j)\}_{j=1}^{J}$, which contains the IDs of voting leaves $\{L_j\}_{j=1}^{J}$ and their matched locations $\{l_k\}_{k=1}^{K}$. Using these two pieces of information it is possible to retrieve all the patches $P_i = (\mathbf{A}_i, \mathbf{s}_i, \mathbf{p}_i, c_i, \mathbf{d}_i)$ associated with the candidate detection $\mathbf{h}$ as follows:

$$E = \{P_i | \forall j, P_i \in L_j, (h_x, h_y) = l_k - \mathbf{d}_i\}. \tag{11}$$

In the following E is referred as support evidences. Each

patch P_i from the support evidences provides a direct link between the candidate detection's location and the corresponding segmentation and pose parameters $(\mathbf{s}_i, \mathbf{p}_i)$. Therefore support evidences are able to cast votes for the candidate's segmentation and pose parameters. We collect all the votes from E into both segmentation and pose subspaces before looking for consensus points as:

$$\hat{c} = \arg\max_{c=1\ldots|E|} \sum_{i=1}^{|E|} \left(\frac{1}{\sqrt{2\pi\sigma_s^2}} \exp\left(-\frac{\|\mathbf{s}_i - \mathbf{s}_c\|^2}{2\sigma_s^2} \right) \right) + \sum_{i=1}^{|E|} \left(\frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_c\|^2}{2\sigma_p^2} \right) \right), \quad (12)$$

which is simply a maximum density estimation in both subspaces in parallel. We fixed σ_s to 0.3 and σ_p to 0.2. The single vote $(\mathbf{s}_{\hat{c}}, \mathbf{p}_{\hat{c}})$ corresponding to locations of highest density in both subspaces is used as the center point for clustering surrounding votes with fixed radius equal to $2 \times \sigma_s$ and $2 \times \sigma_p$ for the segmentation and pose clusters respectively. The clusters' means are updated and their densities are computed to serve as feedbacks for eventually discarding the detection. Candidate detections having weak consensus (clusters of weak densities) in at least one of the two subspaces are discarded. The threshold densities under which detections are discarded have been determined using the validation set from the TUD Multiview dataset. We kept the thresholds which provided the highest performances at EER on the validation set, that is, 7.0 for both the segmentation and pose density thresholds.

The Consensus points are two locations in segmentation and pose subspaces and thus correspond to a unique pedestrian segmentation and pose. We produce full body segmentation and full body pose estimation by back-projecting the consensus points according to the Eq. (10). The detections' corresponding bounding boxes are determined by using the back-projected segmentation masks' boundaries.

4. Experiments

Our experiments are divided into 4 parts, first we compared our detector performances against the original HF [11] on two multiple views pedestrian datasets. Then we compared our detection performances against two Hough based detectors which makes use of a hypothesis verification step. These are the ISM [4] and the recent HF detector from [12]. Then we verified that our detector preserve its high robustness to occlusion in comparison to both the original HF and the popular DPM from [2]. Finally we made a quantitative evaluation of the segmentation mask and pose estimation produced by our pedestrian detector.

4.1 Multi-View Pedestrian Detections

We first compared our Conditional Hough Forest (CHF) against the original Hough Forest [11] on two challenging multiple views pedestrian datasets, namely the PennFudan pedestrian dataset [18] and the test set of the TUD multiview pedestrians dataset [14]. For both datasets, we trained our forest using the 842 pedestrian images described in Sect. 3.3 as our positive samples and used the negative images from the INRIA pedestrian dataset [16] as negative samples. All our segmentation and pose annotations have been projected into the learned subspaces before growing the forests.

Each of the 8 viewpoints was used to train 5 trees resulting in an original forest of 40 trees. A single round of bootstrapping was performed on the negative images set to collect strong false positives. The cropped false positives were used as negative samples to train 5 new trees for each viewpoint. The resulting forest is thus made of 80 trees. Training patches of size 24x24 were extracted both at the current scale and at half the current resolution. 16 patches have been extracted on each positive and each negative training images. Each single patch is described using the features proposed in [1] with blocks of size 8x8 pixels. The resulting feature vectors are then 3x3 arrays of 32 dimensions vectors for each of the two scales. For a fair comparison, the HF code provided by [11] was modified such as to use the same features as our CHF. Multi-scale detection was performed by running the detectors over 6 scales evenly spaced ranging from 1.0 to 0.5. Detections are considered true when their bounding boxes overlap with the ground truth bounding boxes by more than 0.5. A non-maximum suppression step is used to discard multiple detections.

Detection performances can be seen in Fig. 5. Equal Error Rate (EER) and Area under the Curve (AuC) are included between parentheses for each method. We can see that the CHF without using hypothesis verification (CHF no hv) is performing better than the Hough Forest on both the TUD multiview and the PennFudan dataset. A visual inspection of the results reveals that the HF return more false positives than our detector, especially on cluttered background. We believe our detector is making good use of contextual information which makes it easier to prevent detection when the direct background is cluttered.

Introducing hypothesis verification (CHF w. hv) significantly improves the results over HF and the CHF without hypothesis verification, both in terms of precision and recall. Performance at EER improves by 10.1% on the TUD multiview and by 8.6% on the PennFudan dataset over the HF. Furthermore, the AuC improves by 16.6% at 92.6% on the TUD multiview dataset and by 15.1% at 91.9% on the PennFudan dataset. These results illustrate the advantages of using contextual information and the tremendous importance of hypothesis verification for Hough based detectors.

4.2 Hypothesis Verification

Next we compared our detector against two Hough based detectors which make use of a hypothesis verification step, the seminal work of Leibe et al. [4] and the recent detector from Wohlhart et al. [12]. Both methods have reported their

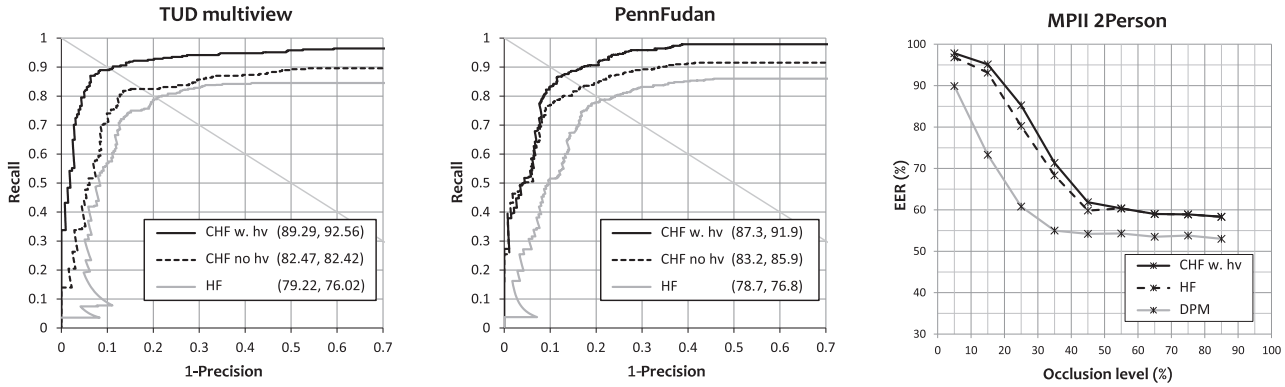


Fig. 5 *(Left and Middle)* Pedestrian detection results on two multi-view pedestrian datasets. EER and AuC performances are shown between parentheses. *(right)* Comparison of detection results under increasingly severe person to person occlusions.

Table 2 Comparative detection results on the TUD pedestrian dataset.

	ISM [4]	HF [11]	HF with hv [12]	**CHF no hv**	**CHF with hv**
EER (%)	80.0	83.6	89.3	85.7	87.7
AuC (%)	83.6	85.9	91.4	87.4	90.9

results on the TUD pedestrian dataset [17] which contains 250 images of side views only, largely un-occluded pedestrians. For fair comparison we re-trained our forest using the 400 side views images available and their flipped version. Once again we train 5 trees per viewpoint and perform a single round of bootstrapping to train additional trees. The final forest contained 20 trees with 10 trees per viewpoint. The subspaces used for hypothesis verification are the same as those used in previous experiments. The detection results are summarized in Table 2.

Once again the CHF with hypothesis verification performs better than CHF without verification steps, which in turn performs better than the HF and the ISM. However the improvements in performances over the HF are relatively less important with 4.1% improvements at EER and 5.0% improvements for the AuC. There are two main reasons for the relatively smaller improvements. First, the training set contains only side views of pedestrians, which means our forest can only votes for side views segmentation and pose. As a results all the votes will be concentrated around those points into both subspaces whether they come from cluttered background or not. Such forest is more likely to produces false detections with high densities of votes in comparison to a forest able to spreads votes over 8 viewpoints. The second reason for the relatively mild improvements is the nature of the dataset. The TUD pedestrian dataset contains relatively mild background clutter which greatly reduces the potential for false detections. With less false detections there is less room for improvements when using the CHF with hypothesis verification.

Comparing with the recent HF [12] we have slightly lower performance at EER 87.7% against 89.3% and nearly identical AuC with 90.9% against 91.4%. Looking at the difference in AuC of only 0.5% it is fair to consider that both detectors perform on par on this dataset. Unfortunately the authors of [12] have only tested their detector on side views datasets of largely un-occluded objects. It is unknown how well their approach works on more challenging datasets with large variations in viewpoints and poses. Moreover their method relies on the presence (activation) of key voting parts whose weights are determined by an SVM at training time. This makes their method very sensitive to the occlusions of some of these key parts. Finally the activation vectors collected to train the SVM are dependent on the forest at hands. This is inconvenient as SVM trained from a given forest's outputs would be of no help to perform hypothesis verification for another forest. This also prevents to add or remove trees to the forest which is necessary for bootstrapping. On a contrary, our forest does not rely on the presence of any key voting parts but solely on the densities of votes within segmentation and pose subspaces. As a result, we retain a high robustness to occlusions, and we can easily add, remove, or re-train trees without affecting the hypothesis verification step.

4.3 Occlusion Robustness

We verified our detector's robustness to occlusions on the MPII-2Person dataset [19] which contains sequences of pedestrian to pedestrian occlusions. There are 9 levels of occlusion ranging from 5% to 85%. Our results in comparison to the HF and the state of the art pedestrian detector DPM [2] can be seen in Fig. 5. Our detector behaves like the HF in the presence of occlusion. Both detectors retaining the advantage over DPM in case of occlusions. A level of occlusion over 45% makes our detector nearly blind to the occluded pedestrian. These results were expected due to the non-maximum suppression step.

4.4 Segmentation and Pose Estimation

Finally we performed quantitative measurements of the seg-

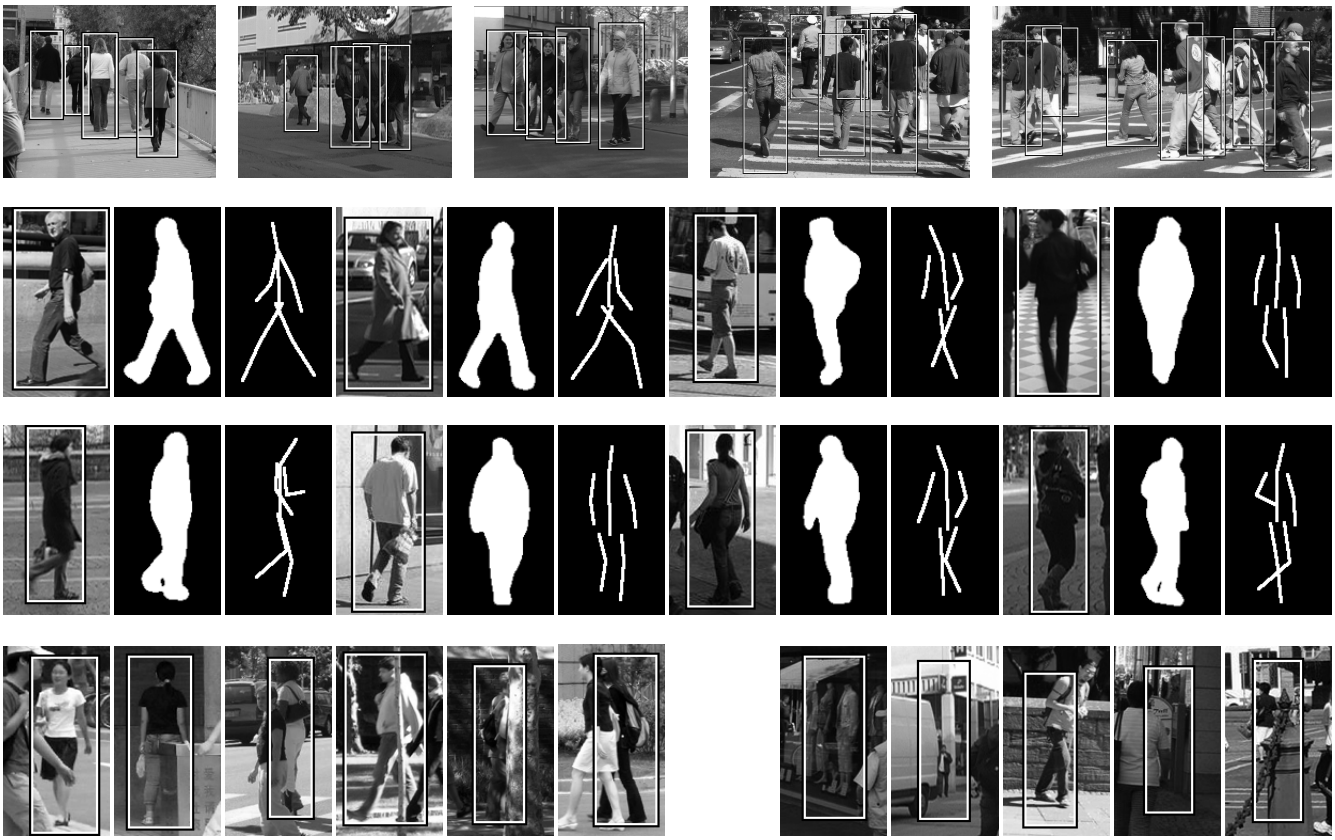


Fig. 6 Multi-views pedestrian detections obtained at EER. *(top row)* Multiple detections with significant occlusions. *(middle rows)* Pedestrian detections and their corresponding estimated segmentations and poses. *(bottom row)* Detections of occluded pedestrians are visible on the left side while false detections are visible on the right side of the bottom row.

mentation and pose estimation performances on the TUD multiview validation set. Some of our detector's results as well as associated segmentations and estimated poses can be seen in Fig. 6. We obtained a segmentation mask precision of 74.46% with a recall of 86.30% which gives us a F-measure equal to 79.94%. For the pose estimation we obtained a mean error in joints position equal to 19.84 pixels with a standard deviation of ±10.32. Our detector often confuses right and left views as well as front and back views which have a big influence on the pose estimation results. When we discard information about right/left limbs, the limbs' mean position error in pixels becomes 13.91 and the standard deviation ±6.66. While both our segmentation and pose estimation are not on par with dedicated segmentation and pose estimation methods, we believe they provide rather good estimations. For example, our segmentation mask would provide a very good initial mask to perform segmentation based Graphcut algorithm. The remaining false detections are typical of pedestrian detectors based on HoG features. It consist of lowly textured vertical structures such as mannequins, poles or human body parts.

5. Conclusion

In this paper, we presented a robust pedestrian detector making use of segmentation and pose cues from training to testing time. We proposed to condition the training of a Hough Forest based on pedestrians' segmentation masks and full body poses. We have introduced a full framework to perform efficient hypothesis verification based on segmentation and pose cues. Our hypothesis verification step not only retain high robustness to occlusion but also provides full body segmentation and pose estimation as byproducts of the detection process. Experiments on 4 challenging pedestrian datasets have shown the significant improvements induced by our hypothesis verification step. We consistently outperformed the baseline Hough forest and performed on par with recent Hough Forest using an inconvenient hypothesis verification step.

More generally, our take-home message is that object detection, segmentation and pose estimation can greatly benefit from each other when combined properly. We hope this work to be a good advocate for research aiming to combine these three fields of research. As a future work, we plan to perform full interaction between the detection, segmentation and pose estimation processes rather than focusing on the detection.

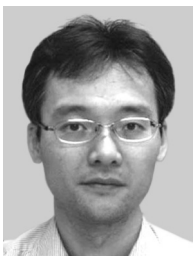
Acknowledgments

This work was supported by "R&D Program for Implementation of Anti-Crime and Anti-Terrorism Technologies for a Safe and Secure Society", Special Coordination Fund for Promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology, the Japanese Government.

References

[1] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," Comput. Vis. Pattern Recognit., pp.1014–1020, 2009.
[2] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," Comput. Vis. Pattern Recognit., pp.1–8, 2008.
[3] Y. Tian, C.L. Zitnick, and S.G. Narasimhan, "Exploring the spatial hierarchy of mixture models for human pose estimation," European Conference on Computer Vision, pp.256–269, 2012.
[4] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," Int. J. Comput. Vis., vol.77, no.1-3, pp.259–289, 2008.
[5] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognition, vol.13, no.2, pp.111–122, 1981.
[6] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," Comput. Vis. Pattern Recognit., pp.2578–2585, 2012.
[7] M. Sun, P. Kohli, and J. Shotton, "Conditional regression forests for human pose estimation," Comput. Vis. Pattern Recognit., pp.3394–3401, 2012.
[8] N. Razavi, J. Gall, P. Kohli, and L. Van Gool, "Latent Hough transform for object detection," European Conference on Computer Vision, pp.312–325, 2012.
[9] M. Sun, G.R. Bradski, B.-X. Xu, and S. Savarese, "Depth-encoded Hough voting for joint object detection and shape recovery, European Conference on Computer Vision, pp.658–671, 2010.
[10] N. Razavi, J. Gall, and L. Van Gool, "Backprojection revisited: Scalable multi-view object detection and similarity metrics for detections," European Conference on Computer Vision, pp.620–633, 2010.
[11] J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," Comput. Vis. Pattern Recognit., pp.1022–1029, 2009.
[12] P. Wohlhart, S. Schulter, M. Köstinger, P. Roth, and H. Bischof, "Discriminative Hough forests for object detection," British Machine Vision Conference, pp.1–11, 2012.
[13] B. Schölkopf, A. Smola, E. Smola, and K-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Computation, vol.10, no.5, pp.1299–1319, 1998.
[14] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3D pose estimation and tracking by detection," Comput. Vis. Pattern Recognit., pp.623–630, 2010.
[15] S. Mika, B. Schölkopf, A. Smola, K. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," Neural Information Processing Systems, pp.536–542, 1999.
[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Comput. Vis. Pattern Recognit., pp.886–893, 2005.
[17] M. Andriluka, S. Roth, and B. Schiele, "People tracking by detection and people detection by tracking," Comput. Vis. Pattern Recognit., pp.1–8, 2008.
[18] L. Wang, J. Shi, G. Song, and I.fan Shen, "Object detection combining recognition and segmentation," Asian Conference on Computer Vision, pp.189–199, 2007.
[19] S. Tang, M. Andriluka, and B. Schiele, "Detection and tracking of occluded people," British Machine Vision Conference, pp.1–11, 2012.

Jarich Vansteenberge received his master degree in electronic systems and multimedia technologies from the Ecole Polytechnique de l'Université de Nantes in 2009. He joined the Kyoto University department of Intelligence Science and Technology as a Ph.D. student in 2010. His research interests includes, face attribute extraction, objects recognition, joint object detection and segmentation as well as human pose estimation.



Masayuki Mukunoki received the bachelor, master and doctoral degrees in information engineering from Kyoto University. He is now an Associate Professor in the Academic Center for Computing and Media Studies and a faculty member in the Graduate School of Informatics, in Kyoto University. His research interests include computer vision, video media processing, lecture video analysis and human activity sensing with camera.

Michihiko Minoh is a professor at Academic Center for Computing and Media Studies (ACCMS), Kyoto University, Japan. He received the B.Eng., M.Eng. and D.Eng. degrees in Information Science from Kyoto University, in 1978, 1980 and 1983, respectively. He served as director of ACCMS from April 2006 to March 2010 and concurrently served as vice director in the Kyoto University Presidents Office from October 2008 to September 2010. Since October 2010, he has been vice-president, chief information officer at Kyoto University, and director-general at Institute for Information Management and Communication, Kyoto University. His research interest includes a variety area of Image Processing, Artificial Intelligence and Multimedia Applications, particularly, model centered framework for the computer system to help visual communication among humans and information media structure for human communication. He is a member of Information Processing Society of Japan, Institute of Electronics, Information and Communication Engineers of Japan, the IEEE Computer Society and Communication Society, and ACM.