A Discriminative Voting Scheme for Object Detection using Hough Forests

Vijay Kumar.B.G
vijay.kumar@elec.qmul.ac.uk
Dr Ioannis Patras
I.Patras@elec.qmul.ac.uk

Multimedia Vision Research Group
Queen Mary, UoL
London, UK

Abstract

Variations of the Implicit Shape Models (ISM) have been extensively used for part-based object detection. Such methods model the information object parts provide about the location of the center and the size of the object in question. Recent object detection techniques employ the generalized Hough transform using random forests, constructing the trees using existing generic criteria for this purpose. In this work, we propose a discriminative criterion for the tree construction, that aims explicitly at maximizing the response at the true object locations in the Hough space while suppressing it at all other locations. To do so, we exploit the knowledge of the object locations in the training images. During training, the Hough images are computed at each node for every training image using the votes from the corresponding training patches. This enables us to utilize a new criterion that discriminates the object locations from the background in the actual Hough space in comparison to the methods that employ classical tree construction criteria. The proposed algorithm results in Hough images with high responses at the object locations and fewer false positives. We present results on several publicly available datasets to demonstrate the effectiveness of the algorithm.

1 Introduction

Numerous approaches have been used for object detection in recent years. Among these, the Implicit Shape Model (ISM) introduced by Leibe [10, 11] forms the basis for many object detection algorithms that use a part-based approach. During training, the ISM learns a model of the spatial occurrence distributions of local patches with respect to anchor points, such as the object center. During testing, this learned model is used to cast probabilistic votes to the location of the center of the object based on the generalized Hough transform technique [5]. Many approaches that use ISM have been proposed [1, 3, 7, 13]. In [3], global shape cues are combined with the local shape cues for pedestrian detection. In [1], boundary fragments are used instead of the appearance of the patches. Malik *et al.* [13] proposed a method that computes weights for the ISM using a max-margin transform. The ISM codebook in [10, 11] is built in an unsupervised way using clustering algorithms. A drawback of such methods is that, matching the patches with the codewords during testing is computationally expensive due to the large number of codewords. To overcome this, Gall *et al.* [7] proposed a Hough forest approach for object detection that employs a tree based approach to learn the patches in a supervised manner. In addition, the computational complexity of matching a patch against a tree is logarithmic to the number of leaves, something that make the approach efficient at runtime.

Random forests introduced by Breiman [9] constitute one of the most popular tree based clustering algorithms known for its simplicity, speed and performance. Random forests use an ensemble of trees for classification and regression tasks. Their computational efficiency was demonstrated by Lepetit *et al*. [12] who used them for realtime keypoint recognition. Moosman *et al*. [14] used extremely randomized trees for building fast discriminative visual codebooks. Schroff *et al*. [15] carried-out segmentation using random forest classifiers.

The Random forests algorithms described above are used primarily for classification problems where the class labels are used for supervised construction of trees and the class information is stored at the leaves of the trees. Gall *et al*. [7] used Random forests for object detection where the class as well as the spatial information of the patches are learned in a supervised manner. The leaves of the trees form a discriminative codebook and store the spatial information along with the class information. This spatial information encodes the location of the center of the object and is used to cast probabilistic Hough votes to the center of the object. Fanelli *et al*. [6] followed a similar Hough forest approach for the localization of the mouth in facial images. In [6, 7], the tree construction process attempts to minimize the class label uncertainty and offset (spatial information) uncertainty towards the leaves. To achieve this, several random tests are performed at each node and the best test that minimizes the offset uncertainty or class label uncertainty is chosen. The class label uncertainty at a node is measured by the entropy of class labels in a node and the offset uncertainty is measured by the distribution of the offsets. Hence, the offset uncertainty measure does not consider the resulting Hough voting space to evaluate tests.

In this paper, we propose an offset uncertainty measure that is defined on the Hough images of the training set. This is in contrast to the Hough forest algorithms in [6, 7] that use a classical regression tree measure, namely the RSS (Residual Sum of Squares) on the leaf nodes, to compute offset uncertainty [17]. In our method, we compute Hough spaces for all the training images when evaluating how good the test split is. The Hough voting spaces are incrementally constructed when building the tree by an algorithm that has a slight overhead (10%) in comparison to the classical one. Our objective is to discriminate the location of the object centers from the background. In order to achieve that we choose the test that has the maximum ratio of votes at objects' center to votes at other locations. The main contribution of the paper is a framework for choosing an optimal test which discriminates the object from background at each node. With this, we achieve application specific Hough forests for object detection that can outperform the general Hough forest technique. The framework also allows us to incorporate different evaluation criterion for training according to the desired output.

The rest of the paper is organized as follows. In Section 2, we give brief introduction of the use of Hough forests for object detection and describe the training procedure. In Section 3, we describe the computation of the intermediate Hough spaces for the training images and the proposed uncertainty criterion using these Hough spaces. In Section 4, we demonstrate the performance of the algorithm on standard datasets and compare it with existing algorithms. Finally, we draw conclusions in Section 5.

2 Overview of Hough Forests

In this section, we describe the Hough forest technique for object detection [6, 7]. The technique uses random forests to model the mapping between the appearance of the patch and its Hough vote. During training, we aim at learning the probability $p(\mathbf{x}|P)$, which encodes the information that a patch with attributes P at location $\mathbf{y}$, provides about the location $\mathbf{x}$ of the object in question. During testing, a patch at location $\mathbf{y}$ casts probabilistic votes $p(\mathbf{x}|P_t(\mathbf{y}),\mathbf{y})$ at the Hough space where a significant local maxima indicates the presence of an object. In what follows, we will summarize the training and voting procedure for Hough forest technique of [7].

2.1 Training

During training, a set of random trees are constructed using the patches of the training images. Let $F = \{f^j\}_{j=1}^{M}$ denote M feature channels of the patch P and f^j is the feature vector of the j^{th} feature channel (e.g. for color images, j = 1 .. 3 are indexes to the color components. Let us denote with $c \in \{0,1\}$ the class label, that is background or foreground and $\mathbf{d} \in \mathbb{R}^2$ the offset between the center of the bounding box and the center of the patch. Once a tree is constructed, any patch P can be propagated through it and end upto one of its leaves, following the path from the root to the leaves according to a test that takes place at each node. The tree expands by performing several random tests at each node and passing the patches to the child nodes based on the result of the tests. The test t for a patch P is given by

$$t(P) = \begin{cases} 0 & \text{if } f^j(\mathbf{p}) < f^j(\mathbf{q}) + \tau, \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{p}$, $\mathbf{q}$ are the coordinates of the randomly chosen indexes in the feature channel f^j and τ is a randomly chosen threshold. The training is performed in a supervised manner by utilizing the class label and offset information of the patches. Once constructed, each leaf node L stores the class information C_L and the offset information $D_L = \{\mathbf{d}\}$ of the patches that end upto it. The class information C_L is the proportion of the object patches (i.e patches with label 1) and D_L is the set of offsets for the patches that end up to the leaf node. Each non-leaf node is assigned a test by choosing the best test from a pool of tests. The tests are chosen such that the uncertainties in class labels and offsets are reduced towards leaves. This is accomplished by using two measures namely the class uncertainty E_c and the offset uncertainty E_o. The class uncertainty measure at a node N is given by

$$E_c(N) = -|N|\big(C_N \log(C_N) + (1-C_N)\log(1-C_N)\big) \tag{2}$$

where C_N is the proportion of object patches at the node N. The offset uncertainty measure is given by

$$E_o(N) = \sum_{\mathbf{d}_i \in D_N} (\mathbf{d}_i - \overline{\mathbf{d}})^2, \tag{3}$$

where $\overline{\mathbf{d}}$ is the mean of the offset vectors in D_N. At each node, the values of $\mathbf{p}$, $\mathbf{q}, \tau$ are randomly chosen for each test and the patches are passed to child nodes according to the test in Eq. 1. The best test among the K tests is chosen as

$$\arg\min_{k} \left[E_*(N_l^k) + E_*(N_r^k)\right] \tag{4}$$

where N_l^k and N_r^k are the child nodes for the k^{th} test and $*$ is chosen randomly between c or o at each node. The construction of a tree stops when either the depth of the tree reaches D_{max} or the number of patches in a node drops below a certain threshold.

2.2 Object Detection

During testing, a number of patches of the input test image located at different locations $\mathbf{y}$ cast probabilistic votes to the location of the object center. Here, the input image is divided into rectangular overlapping patches which are passed through ensemble of trees. Let $P_t(\mathbf{y})$ denote a test patch with center $\mathbf{y}$ which ends up in a leaf node L. It uses the class information C_L and the offset information D_L stored during training to cast votes to the center of the object. The probabilistic votes casted by the patch $P_t(\mathbf{y})$ to the center of the object $\mathbf{x}$ are given by

$$p(\mathbf{x}|P_t(\mathbf{y}),\mathbf{y}) = \left[\frac{1}{|D_L|}\sum_{\mathbf{d}\in D_L}\frac{1}{2\pi\sigma^2}\exp\left(-\frac{\|(\mathbf{y}-\mathbf{x})-\mathbf{d}\|^2}{2\sigma^2}\right)\right]C_L \tag{5}$$

The votes for the forest are obtained by averaging $p(\mathbf{x}|P_t(\mathbf{y}),\mathbf{y})$ over all trees. The final Hough image is computed by combining the votes from all patches for the image in question. The hypothesis is obtained by choosing local maxima in the Hough image that have responses above a certain threshold.

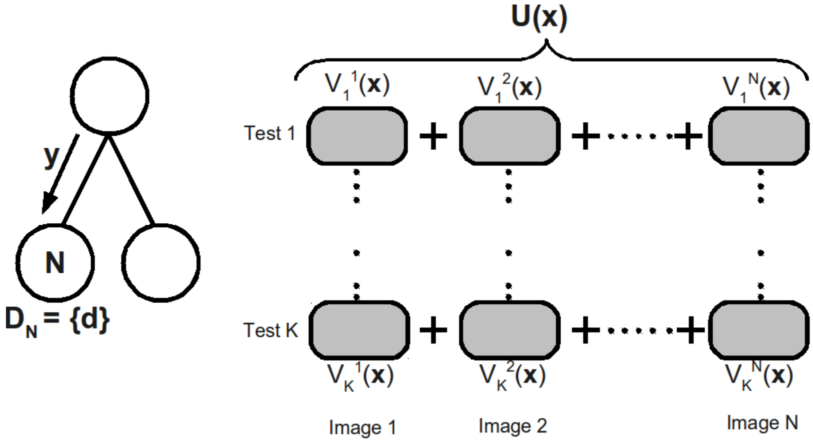

Figure 1: Computation of intermediate Hough spaces at node N

3 Hough Spaces

In this section we first discuss the construction of intermediate Hough spaces and propose a new offset uncertainty criteria for choosing the test at a node based on the constructed Hough spaces. The method in [7] follows a classical tree construction approach by calculating the variance of offsets in the child nodes for the computation of the offset uncertainty measure. This approach tends to reduce the impurity of the offset vectors towards the leaf nodes but it does not consider the actual Hough space. Since the object center location is known for training set images, it can be utilized to verify the predicted object centers which in turn, can be used to evaluate the quality of the tests during training. In this paper, we propose an offset uncertainty measure by determining intermediate Hough spaces for training images at each node for each test and choose a test that maximizes the ratio of the votes at the center of the objects over the votes at other locations.

3.1 Computation of Intermediate Hough Spaces

The value at a pixel $\mathbf{x}$ of the Hough image is computed by accumulating the votes from all the patches in the original image and averaging over all the trees. To compute the Hough image, the whole procedure needs to be repeated for all the pixels in the Hough image, a procedure that is computationally expensive. Alternatively, the Hough space can also be computed by passing a patch $P(\mathbf{y})$ through the trees to determine a leaf node L and adding $\frac{C_L}{|D_L|}$ to the location $\{\mathbf{y}-\mathbf{d}|\mathbf{d} \in D_L\}$. Then smoothing with a Gaussian kernel gives the result of Eq. 5

Here we compute intermediate Hough spaces at each node. More specifically, for each non-leaf node N we calculate the parameters C_N and D_N where C_N is the proportion of the object patches in N and $D_N = \{\mathbf{d}\}$ is the set of offsets in N. The object patches arriving at node N cast votes to the Hough space using the set of offsets in N. This is demonstrated in Fig. 1. Let $V_i^j(\mathbf{x})$ denote the intermediate Hough space for the j^{th} training image and i denote the test index. Let $P(\mathbf{y}^j)$ be a patch from the j^{th} training image arriving at node N. This patch casts votes to the j^{th} Hough image $V^j(\mathbf{x})$ at locations $\mathbf{x} = \mathbf{y} - \mathbf{d}$ where $\mathbf{d} \in D_N$. This procedure is repeated for all the patches arriving at N.

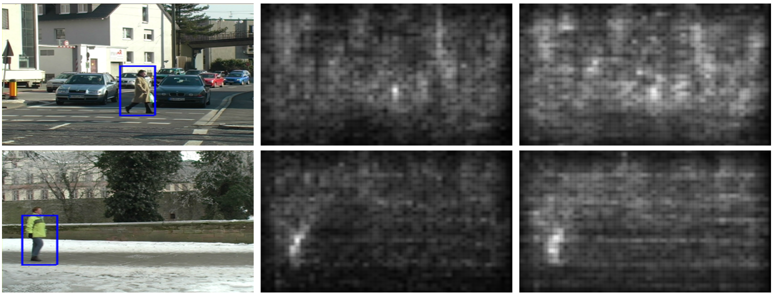

Figure 2: Output image (first), Hough space for the proposed method (middle)[max values: 0.289, 0.299], Hough space for [7] (last) [max values: 0.222, 0.222]

3.2 Proposed Offset Uncertainty Criteria

The Hough forest tree construction process is governed by Eq. 2 and Eq. 3. The class uncertainty criteria attempts to reduce the uncertainties in the class labels towards leaves. The offset uncertainty criteria attempts to reduce the uncertainty in the location of the casted votes. In this work, we propose an offset uncertainty criteria which attempts to maximize the votes at the center of the object locations in the combined Hough image while suppressing responses at other locations. The combined Hough votes for the k^{th} test are computed by considering the votes from the intermediate Hough spaces of all the training images, $\sum_{\mathbf{x}} V_k^j(\mathbf{x})$. The objective is to discriminate the location of the object from the background. Let $\mathbf{x}_c^j$ be the location of the center of the object in the j^{th} Hough image. Let $\mathbf{X}_c^j$ denote an area with few pixels around the center $\mathbf{x}_c^j$ of the object. The proposed offset uncertainty criteria E_{ho} is then given by

$$E_{ho} = \frac{\sum_{j=1}^{T} \sum_{\mathbf{x} \in \mathbf{X}_c^j} V(\mathbf{x})}{\sum_{j=1}^{T} \sum_{\mathbf{x} \neq \mathbf{X}_c^j} V(\mathbf{x})} \tag{6}$$

where at the denominator is the accumulated votes at other parts of all the training images. The above uncertainty criteria is computed for the left and right child nodes of a node and the test that maximizes E_{ho} is chosen at the node under consideration. More specifically, we use the class label uncertainty (Eq. 2) along with the proposed offset uncertainty criterion (Eq. 6) to construct the trees. At each node, a test is chosen from a pool of tests by evaluating the criterion

$$T_1 : \qquad \arg\min_k \left[E_c(N_l^k) + E_c(N_r^k)\right] \tag{7}$$

or

$$T_2 : \qquad \arg\max_k \left[E_{ho}(N_l^k) + E_{ho}(N_r^k)\right] \tag{8}$$

where N_l^k and N_r^k are the left and right child nodes for k^{th} test and a test T_1 or T_2 is chosen randomly. Similar to the approach discussed in Section 2.2, during testing we detect objects at local maxima of the Hough image that are above a threshold.

The proposed uncertainty criterion is based on the response in the output space. Further, additional object specific constraints can be imposed to evaluate the tests thereby improving the performance of the forest. In contrast to the classical offset uncertainty measure (Eq. 3) which ignores the class information, the proposed uncertainty measure utilizes the class information C_N at a node to compute the intermediate Hough space. Since the centers and offsets of the patches for the training images are known apriori, the location of the resulting votes can be pre-computed. This reduces the computational cost during training and significantly improves the tree construction speed.

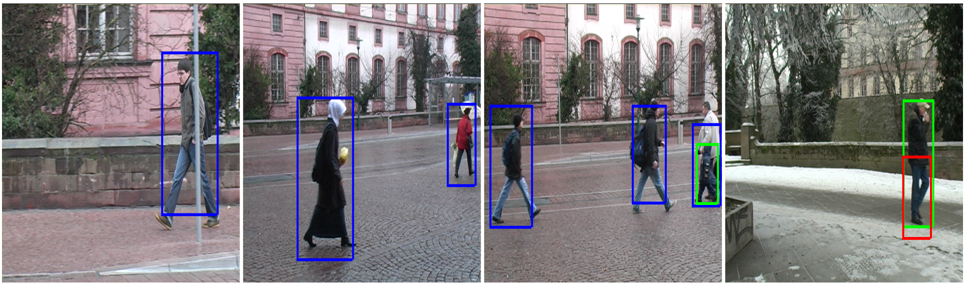

Figure 3: Detected objects (blue), miss detection (green), false positives (red) for TUD dataset

The proposed method computes the Hough space for each test at each node and hence can be computationally expensive. In order to reduce the computational complexity, we observe that the training patches arriving at a node in question cast votes using the offsets of the training patches at the node in question. Since the patch centers and offsets for training patches are known apriori, the voting location for each patch-offset combination can be pre-computed. By pre-computing the voting locations, the computation time is significantly reduced. The proposed method requires 10% more computation time as compared to the classical Hough forests.

4 Experimental Results

We evaluated the performance of the proposed method on three datasets, namely the UIUC-cars, the TUD-pedestrians and the INRIA-pedestrians dataset. In all cases we used 25000 positive and 25000 negative patches for training. We set the maximum depth of the trees to 20 and the threshold for the number of examples/patches in a leaf node to 20. This means that a node is not split if either the maximum depth is reached, or if the number of examples/patches in the node in question are below the threshold. During tree construction, we evaluated 1500 tests at each node and in each case selected the one that minimizes the proposed criterion. We constructed 15 trees for each forest in each dataset and followed a boosting approach to learn hard examples, in which the training examples/patches for a given tree are selected based on the classification result that is obtained using the previously constructed trees. More specifically, hard examples (i.e. misclassified patches) are duplicated in the initial training set. We repeat this procedure every other five trees. For all datasets we consider the size of the bounding box fixed and deal with scale variations by resizing each test image to a number of scales (4 in our experiments), each one of which we pass through the forest in order to obtain a spatial Hough space. Then, a meanshift algorithm is applied in the 3D scalespace in order to obtain hypotheses about the location and scale of the objects.

For the TUD and INRIA datasets, we used 16 feature channels namely the RGB color channels, the magnitude of the horizontal and vertical gradients, the magnitude of second-order derivatives in both horizontal and vertical directions, and the 9 HOG channel descriptors. The HOG descriptors were obtained by accumulating the normalized magnitude in 9 orientation directions computed over a 5×5 window centered around each pixel. To handle variations in clothing, illumination, articulation of parts, the channels were filtered with a max-filter on 5×5 windows centered around a pixel.

We first present results on the UIUC-cars dataset. The training set for this dataset consists of 550 positive examples of images depicting side views of cars and 500 negative images each of size 100×40 pixels. To construct the trees we used 500 positive and 400 negative images from the dataset. We used the three feature channels, namely the gray level values and the absolute values of the horizontal and the vertical gradients. The test set consists of 170 images of the objects with same scale. The criterion suggested by the publisher was used for evaluation. The proposed method achieves 98.5% Equal Error Rate (EER) for the *UIUCSingle*, and therefore performs better than our implementation of the Hough forest [7] which achieves 98% EER for *UIUCSingle* and is in par with the state-of-the-art methods which achieve the same EER Lampert *et al.* [8].

For the same dataset, we demonstrate the effect of width of the area around the ground truth location of the object center of the intermediate Hough spaces $\mathbf{x}_c$ on the final Hough image. $\mathbf{X}_c$ in Eq. 6 denotes an area of a few pixels around the actual center of the object. To determine the effect of width of $\mathbf{X}_c$ on the final Hough image, we constructed Hough forests by varying the size of the area X_c. In Fig. 4 we present the final Hough images for $\|X_{c_1}\| < \|X_{c_2}\| < \|X_{c_3}\|$ for three test images. It can be seen in the figure that the Hough image that are obtained using a small area size (i.e. $\|X_{c_1}\|$) produces high responses at the object locations in the final Hough image and low responses (i.e. small ambiguity/spread) at other locations. As the area increases from $\|\mathbf{X}_{c_1}\| = 52$ to $\|\mathbf{X}_{c_3}\|$ the response at the object location

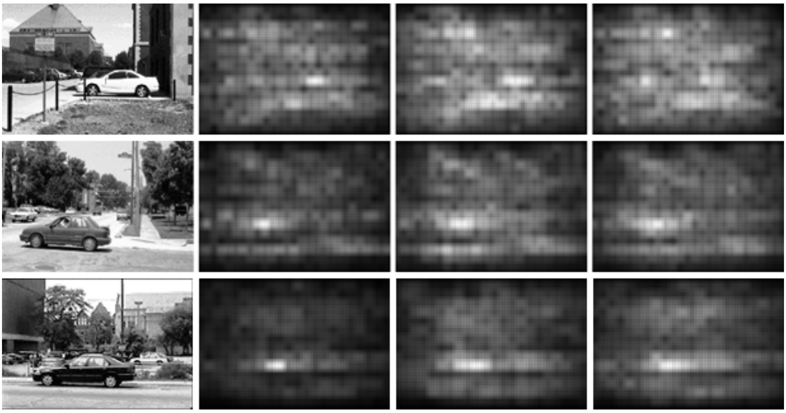

Figure 4: First column: Test images. Second, third and fourth columns: Hough image for $|\mathbf{X}_{c1}| = 52$, $|\mathbf{X}_{c2}| = 370$ and $|\mathbf{X}_{c3}| = 862$ respectively

decrease and the ambiguity in the location of the center increases. The average values of the maximum responses over all training set images are $0.0192, 0.0188$ and 0.0184 for $\mathbf{X}_{c_1}$, $\mathbf{X}_{c_2}$ and $\mathbf{X}_{c_3}$ respectively while the average responses at other areas are $0.0061, 0.0063$ and 0.0065 respectively. In all of our experiments we choose $|\mathbf{X}_{c1}| = 52$, a value that defines an area equal to the 13/100 of the size of the object bounding box.

We then demonstrate the performance of the algorithm on the more challenging TUD-pedestrian dataset [2]. The training set for this dataset consists of 400 images with pedestrians. We trained the Hough forest using positive examples that were constructed by extracting the object bounding boxes and resizing them to fixed dimensions. We constructed 400 negative examples/images by randomly sampling patches from the background areas in the images. The test set consists of 250 images containing 311 fully visible people who exhibit significant variation in clothing and articulation. The evaluation criteria used to measured the detection quality are, cover, overlap and relative distance. Cover and overlap measure how much the ground truth bounding box is covered by the detected bounding box and vice versa. Relative distance measures the distance between the center of the bounding boxes. We inscribe an ellipse in the ground truth bounding box and relate the measured distance to the radious of the ellipse at the corresponding angle [16]. In our experiments, only hypothesis that have cover and overlap more that 50% and relative distance less that 0.5 are accepted as the correct detection.

Fig. 2 depicts the Hough space obtained with our own implementation of the approach in [7] and of the proposed method. It is clear that with the proposed method the number of votes at the center are significantly higher than the number of votes at the background, a fact that indicates higher discrimination capability. Fig. 5 compares quantitatively the performance of the proposed method and the Hough forest algorithm [7]. The proposed method clearly outperforms [7], at high precision values and exhibits similar performance elsewhere. In particular, the proposed method shows 7% improvement over Hough forests at a precision value of 0.93. From the recall-precision curve, it is evident that the proposed method outperforms the Hough forest algorithm for the TUD-pedestrian dataset.

We next demonstrate the performance of the algorithm on INRIA-pedestrian dataset [4]. We used 2416 positive and 2416 negative images for training. The negative images were obtained by sampling the person-free training images. The test set consists of 288 cropped and pre-scaled images with objects and 453 images without the object. Fig. 5 shows the recall vs False Positives per Window curves for the proposed and Hough forest algorithms. It is clear that the propose method outperforms the classical Hough forests particularly at low false positives per window.

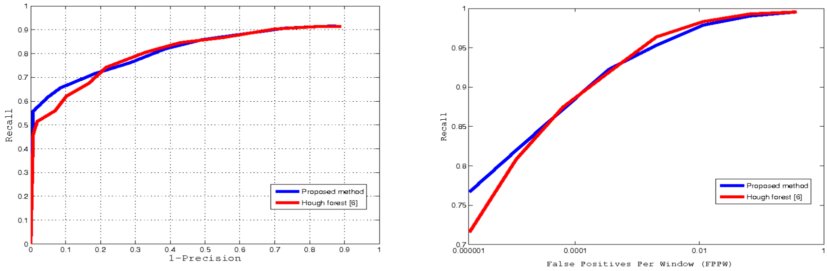

Figure 5: Recall-precision curves for TUD (first column) and INRIA (second column)

5 Conclusion

We introduced a novel framework for object detection using Hough forests that uses the knowledge of the objects' center locations in training images to efficiently construct the trees. During training, we compute at each node the Hough space for each of the training images and use this Hough space to evaluate the tests at the node in question. The test with maximum ratio of votes at the center to the other parts of the combined Hough space is chosen as the candidate test for that node. In that way, we obtain Hough spaces that succesfully discriminate the object from the background. We demonstrated the efficiency of the proposed algorithm through several experiments. We tested the novel framework on standard datasets and have experimentally shown that it outperforms the classical Hough forests.

Acknowledgements

This work has been funded by the EPSRC project "Recognition and Localisation of Human Actions in Image Sequences" (EPG0339351).

References

[1] A. Pinz A. Opelt and A. Zisserman. A boundary-fragment-model for object detection. In *European Conf. on Computer Vision*, 2006.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Int'l Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[3] E. Seemann B. Leibe and B. Schiele. Pedestrian detection in crowded scenes. In *Int'l Conf. Computer Vision and Pattern Recognition*, 2005.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Int'l Conf. Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[5] D.H.Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[6] G. Fanelli, J. Gall, and L. Van Gool. Hough transform-based mouth localization for audio-visual speech recognition. In *British Machine Vision Conference*, September 2009.

[7] Jürgen Gall and Victor Lempitsky. Class-specific Hough forests for object detection. In *Int'l Conf. Computer Vision and Pattern Recognition*, pages 1022–1029, 2009.

[8] Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Int'l Conf. Computer Vision and Pattern Recognition*, volume 0, pages 1–8, 2008.

[9] L.Breiman. Random forests. In *Machine Learning*, September 2001.

[10] Bastian Leibe and Bernt Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference*, pages 9–11, 2003.

[11] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*, pages 17–32, 2004.

[12] Vincent Lepetit, Pascal Lagger, and Pascal Fua. Randomized trees for real-time keypoint recognition. In *Int'l Conf. Computer Vision and Pattern Recognition*, pages 775–781, 2005.

[13] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *Int'l Conf. Computer Vision and Pattern Recognition*, 2009.

[14] Frank Moosmann, Bill Triggs, and Frederic Jurie. Fast discriminative visual codebooks using randomized clustering forests. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, pages 985–992. 2007.

[15] F. Schroff, A. Criminisi, and A. Zisserman. Object class segmentation using random forests. In *British Machine Vision Conference*, 2008.

[16] Edgar Seemann, Bastian Leibe, Krystian Mikolajczyk, and Bernt Schiele. An evaluation of local shape-based features for pedestrian detection. In *British Machine Vision Conference*, 2005.

[17] David S. Vogel, Ognian Asparouhov, and Tobias Scheffer. Scalable look-ahead linear regression trees. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 757–764, 2007.