Object Detection using a Max-Margin Hough Transform *

Subhransu Maji, Jitendra Malik
Computer Science Division, EECS
University of California at Berkeley
{smaji,malik}@cs.berkeley.edu

Abstract

We present a discriminative Hough transform based object detector where each local part casts a weighted vote for the possible locations of the object center. We show that the weights can be learned in a max-margin framework which directly optimizes the classification performance. The discriminative training takes into account both the codebook appearance and the spatial distribution of its position with respect to the object center to derive its importance. On various datasets we show that the discriminative training improves the Hough detector. Combined with a verification step using a SVM based classifier, our approach achieves a detection rate of 91.9% *at* 0.3 *false positives per image on the ETHZ shape dataset, a significant improvement over the state of the art, while running the verification step on at least an order of magnitude fewer windows than in a sliding window approach.*

1. Introduction

Various techniques for object detection have been proposed in the literature including sliding window classifiers, pictorial structures [7], constellation models [8] and implicit shape models [18]. Sliding window classifiers are especially well suited for rigid objects and have been used widely for detection of faces [23, 25], pedestrians [5, 20], cars [24], etc. A binary classifier is evaluated on a uniform sample of the possible locations and scales and is followed by postprocessing step like non-max suppression to find the objects. Some of the popular techniques to alleviate the complexity issue include looking at salient regions, coarse to fine search, branch-and-bound [16].

The Hough transform [6] provides yet another way of dealing with the complexity issue of searching over pose and has been used for various pose estimation problems including shape detection [2]. Of particular interest is the implicit shape model [18] which is a probabilistic formulation of the Hough transform where local parts probabilistically vote for locations of the objects. Combined with verification step, this approach has been used successfully for detection of objects [18, 22]. By allowing the local parts to vote for possible transformations of the object like translation, scale and aspect variation, one can use the peaks of the voting space for importance sampling of windows for further evaluation. Any technique that causes the voting space to better reflect the presence of the object has a direct impact on the speed and accuracy of this two stage classifier.

The main contribution of this paper is to place the Hough transform in a discriminative framework where each local part casts a weighted vote for the possible locations of the object center. The learning framework takes into account both the appearance of the part and the spatial distribution of its position with respect to the object center and parts which are both repeatable and occur at a consistent location are assigned higher weights. The final formulation turns out to be convex and one can obtain a globally optimal solution using off the shelf optimization packages. We call our approach 'Max-Margin Hough Transform' or M^{2}HT. The framework treats the parts (or codewords) and the probability distribution of the object locations as a blackbox, and hence it can be used to learn weights for the popular implicit shape model.

We present experiments on various datasets to show the power of discriminative training by comparing it with a Hough transform that uses uniform codeword weights as well as a simple scheme which we refer to as naive-bayes weights, that takes into account only the "representativeness" of the part and ignores its spatial distribution. On the ETHZ shape dataset [9] the M^{2}HT detector has a detection rate of 60.9% at 1.0 false positives per image compared to 52.4% using uniform weights and 54.2% using naive-bayes weights. On UIUC cars dataset [1] the M^{2}HT detector has half the false positives per image rate at 90% recall compared to the Hough detector based on both uniform and naive-bayes weights. The performance of M^{2}HT is also better than both on the INRIA horses dataset.

*This work is funded by ARO MURI W911NF-06-1-0076 and ONR MURI N00014-06-1-0734

We present further experiments by combining the Hough detector with a verification step using a standard SVM classifier, which then finds the location of the objects by performing a local search around the proposed regions. Our two stage classifier achieves a detection rate of 91.9% at 0.3 false positive per image (FPPI) on the ETHZ shape dataset, a significant improvement over the state of the art, while running the verification step on at least an order of magnitude fewer windows than in a sliding window approach. On UIUC cars we obtain a performace of 97.5% at equal error rate, while having to run the verification step on only 10 windows per image. On INRIA horse dataset the overall detector has a recall of 85.27% at 1.0 FPPI, almost the same as sliding window detector while again considering only a small set of windows per image.

The rest of the paper is structured as follows: We present an overview of the probabilistic Hough transform in Section 2. In Section 3 we cast the voting process in a discriminative framework and outline the max-margin formulation of the problem. The overall detection strategy is described in Section 4. We present our experiments on various datasets in Section 5 and conclude in Section 6.

2. Probabilistic Hough Transform

Let f_i denote the feature observed at a location l_i, which could be based on the properties of the local patch around l_i. Let $S(\mathcal{O}, x)$ denote the score of object $\mathcal{O}$ at a location x. Here x denotes pose related properties such as position, scale, and aspect ratio. Let C_i denotes the i'th codebook entry of the vector quantized space of features f. The implicit shape model [18] framework obtains the overall score $S(\mathcal{O}, x)$ by adding up the individual probabilities $p(\mathcal{O}, x, f_j, l_j)$ over all observations , i.e.

$$S(\mathcal{O}, x) = \sum_j p(\mathcal{O}, x, f_j, l_j) \quad (1)$$

$$= \sum_j p(f_j, l_j) p(\mathcal{O}, x | f_j, l_j) \quad (2)$$

Assuming a uniform prior over features and locations and marginalizing over the codebook entries we get :

$$S(\mathcal{O}, x) \propto \sum_j p(\mathcal{O}, x | f_j, l_j) \quad (3)$$

$$= \sum_{i,j} p(C_i | f_j, l_j) p(\mathcal{O}, x | C_i, f_j, l_j) \quad (4)$$

One can simplify this further using the fact that $p(C_i|f_j, l_j) = p(C_i|f_j)$ because the codebook entries are matched based on appearance only and the distribution $p(\mathcal{O}, x|C_i, l_j, f_j)$ depends only on the matched codebook entry C_i and l_j.

$$S(\mathcal{O}, x) \propto \sum_{i,j} p(C_i | f_j) p(\mathcal{O}, x | C_i, l_j) \quad (5)$$

$$= \sum_{i,j} p(C_i | f_j) p(x | \mathcal{O}, C_i, l_j) p(\mathcal{O} | C_i, l_j) \quad (6)$$

The first term is the likelihood that the codebook entry C_i generated the feature f_j. We base this on the distance of the codebook entry to the feature as follows:

$$p(C_i|f) = \begin{cases} \frac{1}{Z} \exp(-\gamma d(C_i, f)) & \text{if } d(C_i, f) \leq t \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Where Z is a constant to make $p(C_i|f)$ a probability distribution and γ, t are positive constants. The second term is the probabilistic Hough vote for the location of the object, which can be estimated during training time by observing the distribution of the locations of the codebook activations relative to the object center. In our experiments we maintain a binned estimate of $p(x|\mathcal{O}, C_i, l_j)$ by discretizing the space of relative locations of the object. The third term is the weight of the codebook entry emphasizing how confident we are that the codebook entry C_i at location l_j matches the object as opposed to background. Assuming that the probability $p(\mathcal{O}|C_i, l)$ is independent of the location (location invariance) we have a simple way of estimating this using both positive and negative examples as follows :

$$p(\mathcal{O}|C_i, l) = p(\mathcal{O}|C_i) \propto \frac{p(C_i|\mathcal{O})}{p(C_i)} \quad (8)$$

Here, $p(C_i|\mathcal{O})$ is the relative frequency of the codebook entry C_i on the object features , while $P(C_i)$ is the relative frequency on both negative and positive training images. We refer to this as naive-bayes weights, as the weight is set independently for each codebook entry. This takes into account only the appearance of the codebook and ignores the spatial distribution of the part relative to the object center. In the next section we present a way to jointly consider both for learning weights on codebooks. Figure 1 illustrates the detection pipeline for the probabilistic Hough Tranform.

3. Max-Margin Hough Transform

The overall procedure in the previous section can be seen as a weighted vote for object locations over all codebook entries C_i. In this section we will show how to learn these weights w_i in a discriminative manner which directly optimizes the classification performance. The key idea is to observe that the score of the $S(\mathcal{O}, x)$ is a linear function of $p(\mathcal{O}|C_i)$ (making the similar location invariance assumption that $p(\mathcal{O}|C_i, l) = p(\mathcal{O}|C_i)$). One can see this readily

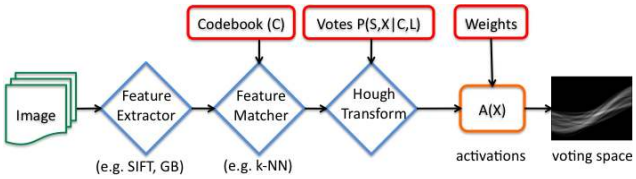


Figure 1. Pipeline for the Probabilistic Hough Tranform (PHT). Given an image local features are extracted which are matched to codebook entries. Votes are cast over possible locations according to a learned distribution over object centers weighted by the importance of each codeword.

from the following :

$$S(\mathcal{O},x) \propto \sum_{i,j} p(x|\mathcal{O},C_i,l_j)p(C_i|f_j)p(\mathcal{O}|C_i,l_j) \quad (9)$$

$$= \sum_{i,j} p(x|\mathcal{O},C_i,l_j)p(C_i|f_j)p(\mathcal{O}|C_i) \quad (10)$$

$$= \sum_{i} p(\mathcal{O}|C_i) \sum_{j} p(x|\mathcal{O},C_i,l_j)p(C_i|f_j) \quad (11)$$

$$= \sum_{i} w_i \times a_i(x) = w^T A(x) \quad (12)$$

where $A^T = [a_1 a_2 \dots a_K]$ is the activation vector and a_i is given by the following equation:

$$a_i(x) = \sum_{j} p(x|\mathcal{O},C_i,l_j)p(C_i|f_j) \quad (13)$$

For a given object location and identity, the summation over j is a constant and is only a function of the observed features, locations and the estimated distribution over the centers for the codebook entry C_i. This suggests a discriminative training algorithm that finds weights that maximize the score S on correct object locations over incorrect ones. Unlike the earlier method of estimating w_i based just on codebook activations, we have the ability to additionally use the conditional distribution of the object centers to learn the weights. In the next section we formalize our training algorithm as well as present experiments to validate our approach.

3.1. Discriminative Training

Let $\{(y_i, x_i)\}_{i=1}^N$ be set of training examples, where $y_i \in \{+1, -1\}$ is the label and x_i is the location of the i'th training instance. Typically we are given the positive instances and pick the "hard" negative instances by finding the peaks in the voting space (using uniform weights) negative training images. The first stage is to compute the activations $A_i = A(x_i)$ for each example by carrying forward the voting process and adding up the votes for each feature f_j found at location l_j according to the Equation 13. Thus the score assigned by the model to the instance i is

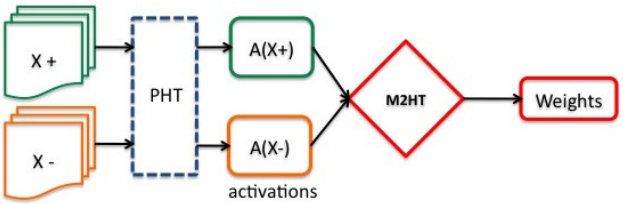


Figure 2. Training pipeline for the "Max-Margin Hough Transform". Given positive and negative training examples with true locations (X±), we obtain the activations A(X±) (Equation 3) for each example, from which weights are learned by using the M^{2}HT learning framework.

$w^T A_i$. Weights are learned by maximizing this score on correct locations of the object over incorrect ones. In order to be robust to outliers and avoid overfitting, we propose a max-margin formulation leading to the following optimization problem :

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{T} \xi_i \quad (14)$$

$$s.t. : \quad y_i(w^T A_i + b) \geq 1 - \xi_i \quad (15)$$

$$w \geq 0, \xi_i \geq 0, \forall i = 1, 2, \dots, N \quad (16)$$

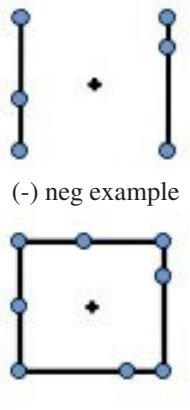
This optimization is similar to the optimization problem of a linear Support Vector Machine [4], with an additional positivity constraint on the weights. We use a standard off the shelf optimization package called `CVX` [13] for solving this problem. Figure 2 shows the training pipeline and Figure 3 shows a toy example which illustrates the differences between the uniform, naive-bayes and M^{2}HT weights.

4. Overall Detection Strategy

The overall detector works in two stages; First the M^{2}HT detector is run on the image and a small set of regions most likely to contain the object of interest is found. Next a verification classifier based on a SVM finds the true location and score of the object by doing a local search around each region by sampling nearby scales and translations. Instead of densely sampling windows all over the image, the Hough step lets us concentrate around the regions most likely to contain the object and at the same time allows us to implicitly sample a wider set of transforms including aspect ratio. We briefly describe the details of both the steps in the next two sections.

4.1. M^{2}HT Detector

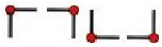
Weights are learned on codebooks generated using k-means clustering of Geometric Blur (GB) features [3] sampled uniformly along the edges in an image. We choose four



(-) neg example

(+) pos example

(a)

						Codebook Weights		
Type	Codebook(C)	$p(C\|-)$	$p(C\|+)$	$A(X-)$	$A(X+)$	UNIF	NB	M^{2}HT
tip		$1/3$	0	1	0	1	0	0
v-edge		$1/3$	$1/4$	$1/5$	$1/5$	1	$6/7$	0
h-edge		0	$1/4$	0	$1/5$	1	2	$1/5$
corner		0	$1/8$	0	1	1	2	1

(b)

Scores	UNIF	NB	M^{2}HT
S(+) = $w^T A(X+)/\|w\|$	$22/40$	$300/380$	$101/105$
S(-) = $w^T A(X-)/\|w\|$	$11/40$	$6/380$	0
$\Delta(S)$ = S(+) − S(-)	$11/40 \approx 0.28$	$294/380 \approx 0.77$	$101/105 \approx 0.96$

(c)

Figure 3. Toy Example. **(a)** Consider the case when we want to detect squares vs. parallel lines. The codebook C, consists of four types of local features: *tip*, *v-edge*, *h-edge* and *corner*. Assume that the tips and corners can localized reliably, but the v/h-edges have an ambiguity in localization as shown by the blue dots in **(a)**. Furthermore for simplicity assume that the a *v/h-edge* feature can be found in one of five discrete locations along the edge, i.e. $p(X + \mid$ *v/h-edge*$) = 1/5$. Table **(b)** shows various steps in the Hough voting. $p(C|\pm)$ is the probability of each codebook type in the $\pm$ example. $A(X\pm)$ are the codebook activations obtained using Equation 3. Naive Bayes(NB) weights $\propto \frac{2p(C|+)}{p(C|+)+p(C|-)}$ correctly ignores the *tip* features and downweighs the *v-edge* features, but the *h-edge* and *corner* features are weighted equally. M^{2}HT on the other hand downweighs the *h-edge* features as they do not localize well. In addition it completely ignores the *v-edge* features as they contribute equally to the $+$ and $-$ example. Table **(c)** shows the scores according to the weights for various schemes. Notice how M^{2}HT weights assigns a score of 0 on the negative example while at the same time achieving a high separation between the $+$ and $-$ example. It finds the corners as the most important feature type.

orientation directions and the outer radius of the GB feature typically as 20% of the object height, giving us a good tradeoff between the repeatability and discriminativeness of the features. In our simple implementation we found that voting over scales is not as reliable, so instead we run the M^{2}HT detector at a small set of scales and vote for the rest of the pose parameters. The choice of local features is arbitrary and we choose local features based on patches of fixed size for simplicity, while taking note that [15] has recently proposed a Hough transform based detector using regions as local features. On the positive set of training images, the relative pose of the center of the object is recorded during training time and a binned approximation of this distribution is maintained. Negative examples are obtained by first running the Hough detector using uniform weights on negative images and finding the peaks in the voting space above a threshold. Negative images are typically ones not containing the category of interest. Activations (A_i) are then computed on the set of positive and negative locations and are used to learn the weights of the codebook entries using our max-margin formulation.

4.2. Verification Classifier

We train a SVM classifier using the pyramid match kernel [14, 17] on histograms of oriented gradients as features. Responses to $[-1\ 0\ 1]$ and $[-1\ 0\ 1]^T$ filters define the gradients from which histograms in 9 orientations are computed. The image is divided into grids of increasing resolutions for 4 levels, and histograms from each level are weighted according to the equation $w_l = 2^{l-1}$, $l = 1$ being the coarsest scale, and concatenated together to from one long feature vector. A SVM is trained using the histogram intersection kernel on these features. We refer to this as the IKSVM classifier. On all datasets training is done by resizing the positive instances of the category to the median aspect ratio and a number of windows sampled from negative training images serve as negative examples. To detect an instance of an object in the sliding window mode the classifier is run at various locations and scales by keeping the aspect ratio of the image fixed. Search over aspect ratio adds another factor to the run time, so we do not do it. A simpler baseline would have been to use a linear kernel, but others [9] have noticed that on the ETHZ shape dataset, linear SVM does not give full recall. We use the speedup method for IKSVM classification proposed in [19] which makes the runtime of the classifier essentially equivalent to a linear SVM.

5. Experimental Results

In all our experiments we would like to verify two things: **(1)** The M^{2}HT detector should have a better performance compared to Hough transform detector using

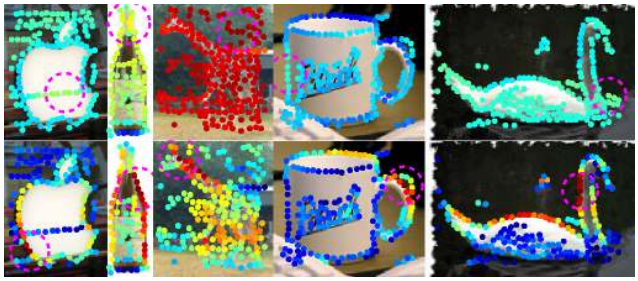
uniform weights or naive-bayes weights. Quantitatively this means a lower false positive rate for the same recall. **(2)** The performance of the two stage M^{2}HT + IKSVM detector should be comparable to the IKSVM detector in the sliding window mode, while having to evaluate the IKSVM detector on a significantly fewer locations. Additionally, if the Hough transform votes for pose parameters like aspect ratio we would like to see that the two stage detector is robust to these pose changes. Finally the overall approach should compare favorably to other approaches in the literature both in terms of accuracy and space-time complexity. To validate our claims we present our experiments on the ETHZ shape, UIUC cars and INRIA horses dataset.

5.1. ETHZ Shape Dataset

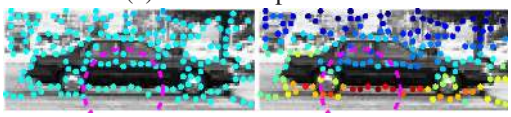
We first report our results on the ETHZ Shape Dataset. It contains 255 test images and features five shape-based classes (apple logos, bottles, giraffes, mugs, and swans). For training we use half the positive examples and an equal number of negative examples equally distributed among the rest of the categories. All the remaining examples are used for testing. We use the same training and test splits used by authors of [9] for a fair comparison.

M^{2}HT Detector Training : For the Hough training step all ground truth bounding boxes of a particular category are scaled to a height of 96 pixels, while keeping the aspect ratio fixed. A separate codebook is learned for each category using k-means clustering with $k = 400$ centers. For categories like mugs and giraffes the aspect ratio varies widely so we train the Hough detector to vote for both the center and aspect ratio of the object. We maintain a binned approximation of distribution of the location of the center with bin width=10px, bin height=10px and aspect width=0.1. We then run the max-margin training procedure described in Section 4.1 to learn the weights for the codebook entries. Figure 4(a), shows the learned weights for various categories. The learning framework assigns high weights to parts of the object which are both characteristic and are good predictors of the object location, while simultaneously ignoring the background clutter in images. Notice that we only use the groundtruth bounding box for training, which includes a significant amount of background clutter for categories like giraffes and swans. The naive-bayes weights are strongly affected by rarely occurring structures in the background. Table 1 shows the detection rates for various weights. The learned weights do better than both uniform and naive-bayes weights.

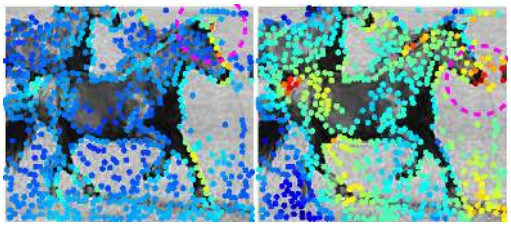
Overall Detector Results : Table 1 shows the results for both the IKSVM detector used in the sliding window mode at a fixed aspect ratio and the M^{2}HT + IKSVM detector. Precision/Recall and Detection Rate/FPPI plots for the M^{2}HT + IKSVM detector are in Figure 5. The IKSVM baseline is quite good and achieves a detection rate



(a) ETHZ Shape Dataset



(b) UIUC Cars



(c) INRIA Horses

Figure 4. Naive Bayes / Learned weights using M^{2}HT, on various categories of the ETHZ shape **(a top / bottom)**, UIUC cars **(b left / right)** and INRIA horses dataset **(c left / right)** respectively. For each category the colors represent the strength of the weights (dark red is highest) and are on the same scale for both naive-bayes and M^{2}HT. In each image, the most important part is circled with radius equal to the outer radius of the *GB* descriptor. Notice how the salient parts like the handles on mugs, the neck and leg regions of the giraffe are assigned high weights, while the background clutter is ignored. The naive-bayes weights are strongly affected by rare structures in the background. On UIUC cars features at the bottom of the car like shadows and wheels are emphasized which are both repeatable and good predictors of the object center, while on INRIA horses regions near the head and tail are found to be most important.

of 87.7% (0.3 FPPI) and 88.48%(0.4 FPPI). Sampling the nearby scales and locations around the regions proposed by the Hough transform leads to an improved detection rate of 91.9%(0.3 FPPI) and 93.2%(0.4 FPPI). Including the windows of the local search is still at least two orders of magnitude fewer than a sliding window detector for a similar dense sampling. Additionally we implicitly sample over aspect ratios because the Hough detector proposes regions of various aspect ratios. This leads to a significant improvement over the baseline IKSVM detector for the giraffe and mugs category, where the aspect ratio varies widely. Our results are significant improvement over previous best results 61.4% of KAS [9] and 67.1% of TPS-RPM [10] at 0.3 FPPI. While the results of TPS-RPM are not directly

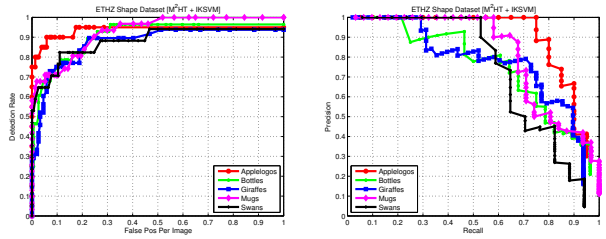


Figure 5. Detection plots using the M^{2}HT + IKSVM on ETHZ shape dataset. **(Left)** Detection Rate vs. FPPI ; **(Right)** Precision vs. Recall. All results are reported using the PASCAL criterion.

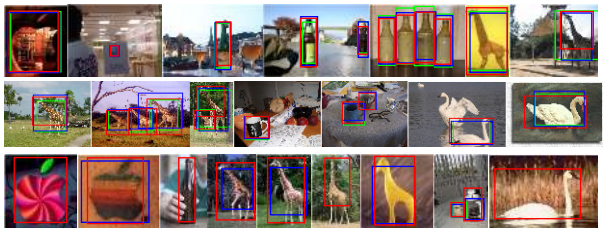


Figure 6. **(Rows 1-2)** Example detections on the ETHZ shape dataset using M^{2}HT+IKSVM detector (blue), IKSVM detector used in sliding window mode (green) overlaid with groundtruth (red). **(Row 3)** Example images where at least one of the two detectors fails.

comparable as the authors report numbers using a five-fold cross validation, ours are still better considering that the average standard deviation in accuracy over trials is about 9%. However [10] has the additional ability to localize the object boundaries. The only other approach that achieves comparable results on this dataset is the contemporaneous work of recognition using regions [15]. Figure 6 shows examples of detections and misses for various categories.

5.2. UIUC Cars

This database was collected at UIUC [1] and contains images of side views of cars. The training set consists of 550 car and 500 non-car images. We test our methods on the single scale image test set which contains 170 images with 200 cars. The images are of different sizes themselves but contain cars of approximately the same scale as in the training images.

M^{2}HT Detector Training : Similar to the ETHZ dataset we compute GB features on both the positive and negative windows by uniformly sampling points on the edges and learn a codebook using k-means with $k = 100$. For every cluster the conditional distribution of the center of the object is maintained as binned approximation with a bin width=4px and bin height=4px. This is a fairly dense sampling given that the training images are 100×40, so we spatially smooth the bins to avoid any artifacts. A second loop over the training images is done and activations are computed and codebook weights are learned using the M^{2}HT framework. Figure 4(b) shows the learned weights using max-margin training and naive bayes. Notice how the features near the bottom of the car are emphasized, which are both repeatable and good predictors of the object center.

M^{2}HT/Overall Detector Results : Figure 7(Left) shows the recall as a function of false positives per image for various learning schemes. At 90% recall the M^{2}HT detector has about half as many false positives per image than the Hough detector using uniform weights or naive bayes weights. Considering only the top 10 windows per image and running the IKSVM verification step leads to recall of 97.5% at equal error rate an improvement of 1.74% over IKSVM detector used in the sliding window mode, while having to consider $10\times$ fewer regions per image (Figure 7(Middle)). The increased precision is because the IKSVM classifier densely samples windows near the most likely locations of the object, while being able to discard a large fraction of the regions in the image not containing an object. Our method compares favorably to other methods in the literature as shown in Figure 7(Right).

5.3. INRIA Horses

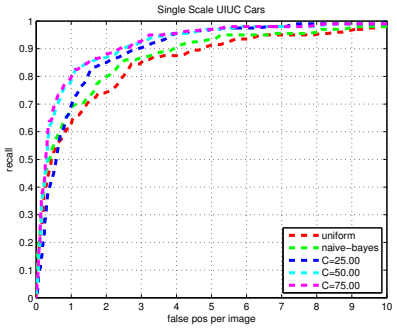
The INRIA horse dataset collected by Jurie and Ferrari, consists of 170 images with one or more side-views of horses and 170 images without horses. Horses appear at several scales, and against cluttered backgrounds. We use the same training and test split of [9] consisting of 50 positive and 50 negative examples for training and the rest for testing.

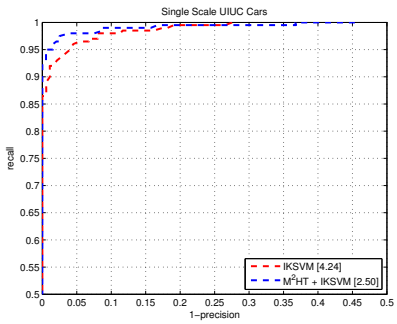
M^{2}HT Detector Training : We learn a codebook using k-means with $k = 500$ and learn weights for each cluster center. Figure 4(c) shows the weights learned for various features using the max-margin training and naive-bayes scheme. The IKSVM classifier is trained by resizing all the ground truth bounding boxes to the median aspect ratio of all horses in this dataset.

M^{2}HT/Overall Detector Results : Figure 9 shows the performance of the M^{2}HT detector and the overall detector. The M^{2}HT detector outperforms both the naive-bayes and the uniform weights. The overall performance of the M^{2}HT + IKSVM detector is same as the IKSVM detector while having to consider only 25 windows per image, which is up to two orders of magnitude fewer than the sliding window classifier. At 1.0 false positive per image we have a detection rate of 85.27% for M^{2}HT + IKSVM and 86% for IKSVM compared to previously published results of 80.77%[9, 11] and 73.75% [10]. The results of [10] are however not directly comparable as the authors report results using 5-fold cross validation. Figure 9(Right) shows some detections and misses on this dataset.

	Hough Detector Recall @ 1.0 FPPI			**Overall Detector** Recall @ 0.3/0.4 FPPI			
Category	UNIF	NB	M^{2}HT	IKSVM	M^{2}HT + IKSVM	KAS[9]	TPS-RPM [10]*
Applelogos	70.0	70.0	85.0	90.0/90.0	95.0/95.0	50.0/60.0	77.7/83.2
Bottles	62.5	71.4	67.0	96.4/96.4	92.9/96.4	92.9/92.9	79.9/81.6
Giraffes	47.1	47.1	55.0	79.1/83.3	89.6/89.6	49.0/51.1	40.0/44.5
Mugs	35.5	35.5	55.0	83.9/83.9	93.6/96.7	67.8/77.4	75.1/80.0
Swans	47.1	47.1	42.5	88.2/88.2	88.2/88.2	47.1/52.4	63.2/70.5
Average	52.4	54.2	60.9	87.5/88.4	91.9/93.2	61.4/66.9	67.1/71.9

Table 1. Performance of various algorithms on the ETHZ shape dataset. All the results are reported using the PASCAL criterion (Intersection/Union ≥ 0.5). The Hough detector based on discriminatively learned weights (M^{2}HT) alone has a detection rate of 60.9% at 1.0 FPPI, an improvement of 6.7% over the naive bayes weights (NB) and 8.5% over uniform weights (UNIF). The IKSVM classifier when used in sliding window mode has a average detection rate of 87.5% at 0.3 FPPI. By combining with the Hough detector, the performance improves to 91.9% at 0.3 FPPI. There are significant improvements in the giraffe and mugs category, which have high variation in aspect ratio. This is a significant improvement over previous best results 61.4% of KAS [9, 11] and 67.1% of TPS-RPM [10] at 0.3 FPPI. *(The results of TPS-RPM are not directly comparable as the authors report numbers using a 5-fold cross validation.)**





Method	**Performance**
IKSVM	95.76 %
M^{2}HT + IKSVM	97.50 %
Agarwal & Roth [1]	79 %
Garg et al. [12]	88 %
Fergus et al. [8]	88.5 %
ISM [18]	97.5 %
Mutch & Lowe [21]	99.6 %
Lampert et al. [16]	98.5 %

Figure 7. **(Left)** Detection plots on UIUC car dataset for various values of the learning parameter C using the max-margin Hough training. At 90% recall the false positive rate is only about half compared to both uniform weights and naive bayes weights. **(Middle)** Combining with the verification step using the IKSVM classifier. Only the top 10 windows per image are considered, which is at least $10\times$ fewer than the number of windows considered by a sliding window detector. By sampling around the regions proposed by the Hough detector there is an improvement of 1.74% over the sliding window detector. **(Right)** Performance at Equal Error Rate on UIUC Single Scale Cars for various methods.

6. Conclusions

The main contribution of this paper is to cast the Hough transform in discriminative framework which leads to better accuracy on various datasets compared to both uniform and naive-bayes weights. The implicit shape model may benefit from this framework to learn the weights as our framework treats both the parts and spatial models as a blackbox. The final problem is convex and easy to optimize using off the shelf optimization packages. The proposed two stage M^{2}HT + IKSVM detector has better runtime complexity than a sliding window detector and at the same time is more robust to pose variations leading to state of the art results on ETHZ shape dataset and competitive results on the UIUC car and INRIA horse dataset.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV (4)*, pages 113–130, 2002.

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

[3] A. C. Berg and J. Malik. Geometric blur for template matching. In *CVPR (1)*, pages 607–614, 2001.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893, 2005.

[6] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR (2)*, pages 264–271, 2003.

[9] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):36–51, 2008.

Figure 8. Example detections(green) and mis-detections(red) using the M^2HT + IKSVM detector on UIUC cars dataset.

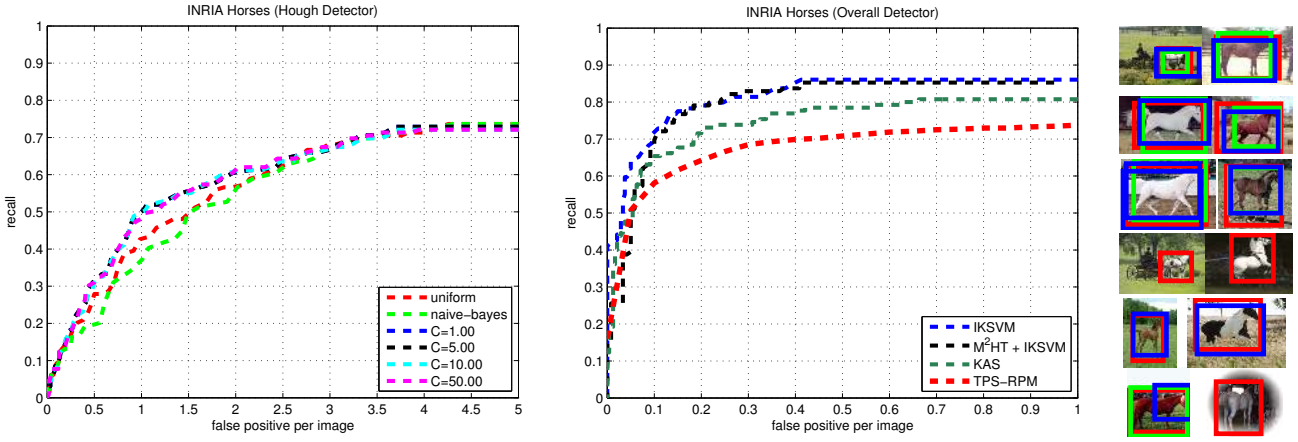


Figure 9. Detection plots on INRIA Horses dataset using the PASCAL criterion. **(Left)** Comparison of M^2HT detector for various choices of the learning parameter C vs. uniform weights and naive-bayes weights. The M^2HT detector consistently outperforms both. **(Middle)** Overall detections results using IKSVM and two stage M^2HT + IKSVM. Performance of M^2HT + IKSVM is similar to IKSVM while having to consider only 25 windows per image on average, which is up to two order of magnitude fewer than in sliding window approach. At 1.0 false positive per image we have a detection rate of 85.27% for M^2HT + IKSVM and 86% for IKSVM compared to previously published results of 80.77% (KAS) [9, 11] and 73.75%(TPS-RPM) [10]. **(Right)** Example detections and misses on the INRIA horse dataset using M^2HT+IKSVM detector (blue), IKSVM detector used in sliding window mode (green), overlaid with groundtruth (red). *(Note that the results of TPS-RPM are not directly comparable as the authors report numbers using 5-fold cross validation.)*

[10] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, 2007.

[11] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. In *Technical Report , INRIA, RR 6600*, july 2008.

[12] A. Garg, S. Agarwal, and T. S. Huang. Fusion of global and local information for object detection. *Pattern Recognition, International Conference on*, 3:30723, 2002.

[13] M. Grant and S. Boyd. Cvx: Matlab software for disciplined convex programming (web page and software). http://stanford.edu/∼ boyd/cvx. April 2008.

[14] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005.

[15] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, 2009.

[16] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008.

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR (2)*, pages 2169–2178, 2006.

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

[19] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

[20] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1863–1868, 2006.

[21] J. Mutch and D. G. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR (1)*, 2006.

[22] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, pages 575–588, 2006.

[23] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *CVPR*, pages 130–136, 1997.

[24] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[25] P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.