On Detection of Multiple Object Instances using Hough Transforms

Olga Barinova Victor Lempitsky Pushmeet Kohli

Abstract—Hough transform based methods for detecting multiple objects use non-maxima suppression or mode-seeking to locate and distinguish peaks in Hough images. Such postprocessing requires tuning of many parameters and is often fragile, especially when objects are located spatially close to each other. In this paper, we develop a new probabilistic framework for object detection which is related to the Hough transform. It shares the simplicity and wide applicability of the Hough transform but at the same time, bypasses the problem of multiple peak identification in Hough images, and permits detection of multiple objects without invoking non-maximum suppression heuristics. Our experiments demonstrate that this method results in a significant improvement in detection accuracy both for the classical task of straight line detection and for a more modern category-level (pedestrian) detection problem.

Index Terms—Hough Transforms, Object Detection in Images, Line Detection, Scene Understanding.

1 Hough Transform in Object Detection

The Hough transform [1] is one of the classical computer vision techniques which dates 50 years back. It was initially suggested as a method for line detection in edge maps of images but was then extended to detect general low-parametric objects such as circles [2]. In recent years, Hough-based methods were successfully adapted to the problem of part-based category-level object detection where they have obtained state-of-the-art results for some popular datasets [3]–[8].

Both the classical Hough transform and its more modern variants proceed by converting the input image into a new representation called the *Hough image* which lives in a domain called the *Hough space* (Figure 1). Each point in the Hough space corresponds to a hypothesis about the object of interest being present in the original image at a particular location and configuration. The dimensionality of the Hough image thus equals the number of degrees of freedom for the configuration(+location) of the object.

Any Hough transform based method essentially works by splitting the input image into a set of *voting elements*. Each such element votes for the hypotheses that might have generated this element. For instance, a feature that fires on faces might vote for the presence of a person's centroid (torso) in location just below it. Of course, voting elements do not provide evidence for the exact localization and thus their votes are distributed over many different hypothesis in the Hough space. Large values of the vote are given to hypotheses that might have generated the voting element with high probability. The votes from different voting elements pixels are added together into a Hough image. The objects of interest are then detected as peaks in the Hough image, with the height of the peak providing the confidence of the detection.

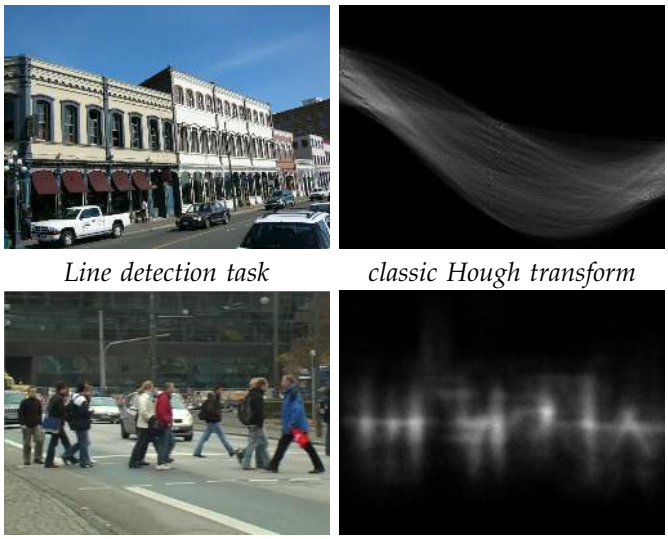


Fig. 1. Variants of Hough transform render the detection tasks (left) into the tasks of peaks identification in Hough images (right). As can be seen, in the presence of multiple close objects identifying the peaks in Hough images is a highly non-trivial problem in itself. The probabilistic framework developed in this paper addresses this problem without invoking non-maxima suppression or mode seeking heuristics.

. *Olga Barinova is with Lomonosov Moscow State University, Moscow, Russia. Email: obarinova@graphics.cs.msu.ru.*
Victor Lempitsky is with Yandex, Moscow, Russia. Email:victorlempitsky@gmail.com.
Pushmeet Kohli is with Microsoft Research, Cambridge, UK. Email:pkohli@microsoft.com

The popularity of the Hough-based approach to object detection stems from its flexibility (e.g. the primary voting elements are not restricted to be edge pixels, but can include interest points [3], image patches [5], [7], or image regions [6]). Another attractive property is the simplicity of the learning procedure. Given a set of images annotated with the location of objects of interest, learning essentially involves construction of the appearance codebook (or voting elements). The Hough vote for each codebook entry is then simply obtained from the training data by observing the distribution of object parameters (e.g. centroid displacements) which generates the entry. This simple additive nature of the Hough transform makes the detection process robust to deformation, imaging noise and many kinds of occlusion.

In spite of all the advantages mentioned above, the Hough transform still has a major flaw in that it lacks a consistent probabilistic model. This leads to problems of both theoretical and practical nature. From the theoretical viewpoint, Hough-based detection does not allow hypotheses to *explain away* the voting elements. To give an example, consider the situation when the maximum in the Hough image corresponds to a correctly detected object. Consider also the voting elements that were generated by this object. These elements are likely to cast strong votes for the detected object, but they are also likely to cast votes for other hypotheses, and the strength of those spurious votes is not inhibited in any way by the fact that a good hypothesis explaining the voting element already exists. In practice, this means that various non-maxima suppression (NMS) heuristics have to be used in real detection scenario to localize peaks in the Hough image. These heuristics typically involve specification and tuning of several parameters.

The goal of the paper is to introduce a new detection method similar to the Hough transform. More precisely, we introduce a new framework, which has a probabilistic nature, and shares most of the virtues of the Hough transform. Notably, the new model can reuse the training procedures and the vote representations developed in previous works on Hough-based detection, such as Implicit Shape models [3] or Hough forests [5]. At the same time, the new approach bears some additional important advantages over the Hough Transform:

- It performs multiple objects detection via an energy optimization (MAP-inference in the probabilistic model), in contrast to heuristic peak location followed by non-maxima suppression used in vanilla Hough Transform.
- Experimental results show that it results in a better accuracy for images containing multiple objects of interest.
- Its probabilistic nature means that our model is easy to integrate with other approaches, including e.g. modeling of higher-level geometric constraints on the location of the objects.

The disadvantage of the suggested approach compared to the traditional Hough transform is the increased computation time, which however, is still very competitive compared to many other modern detection techniques.

The remainder of the paper is organized as follows. We start by reviewing the Hough transform from the probabilistic viewpoint and introducing our model in section 2. We then discuss how MAP-inference in our model can be performed. We proceed to the experimental comparison of our model with the vanilla Hough transform method. This evaluation is performed on the traditional Hough task of line detection in images, as well as the more modern task of category-level (pedestrian) detection. Finally, we discuss the relation of our approach to prior art and how our framework can be extended and integrated with other approaches.

2 Related work

The model for Hough voting presented in this paper is related to a number of existing frameworks for model fitting, object recognition, and image segmentation.

The problem of developing a probabilistic interpretation of the Hough transform has been considered by works in the literature. Stephen [9] was one of the first to address this issue by allowing features (or image elements) to be generated from the object or from an outlier process. Minka [10] tried to explain why summing the likelihoods (used in conventional Hough transform) perform better than multiplying them. He did this by showing that summation leads to an 'outlier' model. Allan and William [11] built on the above mentioned works by using the Hough transform for object localization.

A number of recent works (Hoiem et al. [12], Lazic et al. [13], Delong et al. [14], and Ladicky et al. [15]) have considered the use of energy functionals with terms corresponding to the presence of particular labels. Our reformulation of Hough transform leads to an energy function, which belongs to the class of energies considered in those works. The application of our approach to line detection in edge maps is thus directly related to geometric fitting frameworks in [13], [14], while our second target application, namely part-based object detection, is similar to the approaches developed in [12], [15]. Our approach links similar probabilistic and energy optimization ideas with the Hough transform paradigm. The fact that our priors do not incur spatial smoothness between image elements (unlike [12], [14], [15]), allows us to retain much of the speed of the traditional Hough transform especially when the number of objects is small, and also to avoid the use of sparsification heuristics (proposal generation) required in all the methods discussed above.

Related to our task of Hough-transform based object detection, Leibe et al. [3] considered detection of multiple objects instances in Hough-based framework by using object segmentation and a MDL prior to prune out false hypothesis. However, their method also involves

reasoning about the segmentation support of individual objects and their overlaps, which makes the probabilistic and energy aspects of their formulation less clear. A recent work of Lehmann et al. [16] also aimed at deriving a probabilistic model for Hough-based object detection. Their reformulation reveals an interesting link between Hough-based and sliding window object detectors but does not address the problem of the detection of multiple object instances. At the same time, Desai et al. [17] demonstrated how non-maximum suppression for sliding window classifiers can be trained discriminatively in a max-margin framework.

Apart from the set of recent works discussed above, it has to be noted that the use of minimum description length (MDL) principle and similar priors in computer vision dates back long ago at least to the segmentation works of Leclerc [18] and Zhu and Yuille [19]. Later on, an approach that optimizes over sets of object instances, treating these sets as competing interpretations of the object parts was presented by Amit et al. in [20].

3 THE FRAMEWORK

3.1 Analysis of the Hough transform.

We start by introducing our notation and then analyze the probabilistic interpretation of Hough-based detection. Let us assume that the image observations come in the form of N *voting elements*, which throughout the paper we will index with the letter i. These elements may correspond to e.g. pixels in the edge map (in the case of line detection), or to interest points (in the case of the implicit shape model-like framework). Our framework allows the use of generic voting elements: pixels, patches or segments, which are sampled densely or sparsely using interest point detectors. Figure 2(a) shows densely sampled patches as voting elements.

We also assume a Hough (or hypothesis) space $\mathcal{H}$, where each point $h \in \mathcal{H}$ corresponds to a hypothesis about the presence of an object of interest (e.g. a line, a pedestrian) in a particular location/configuration. Figure 2(a) shows the bounding boxes corresponding to some candidate object detection hypotheses.

The detection task can then be formulated as finding the finite subset of the Hough space that corresponds to objects that are actually present in the image. To formalize this, for each hypothesis $h \in \mathcal{H}$, we introduce the binary random variable y_h that takes the value 1 if the hypothesis actually corresponds to a real object and the value 0 otherwise.

The Hough transform does the detection by considering each voting element i independently and reasoning which object h might have generated it. To formalize this reasoning, we introduce a random variable x_i that takes a value in the augmented Hough space $\mathcal{H}' = \mathcal{H} \cup 0$. The assignment $x_i = h \in \mathcal{H}$ implies that the voting element i is generated by the object h, while $x_i = 0$ implies that element i comes from background clutter and is not part of any object of interest. We can now consider *votes* as (pseudo)-densities $V(x_i = h|I_i)$ in the Hough space conditioned on the descriptor I_i of the voting element i. The descriptor here might include the geometric position of the voting element in the (scale)space and/or the local image appearance. These conditional (pseudo)-densities are then added and the peaks of the resulting sum are considered as valid hypothesis.

The summation operation within the Hough transform tacitly assumes that pseudodensities correspond to the logarithms of the object likelyhoods $V(x_i = h|I_i) = \log p(x_i = h|I_i)$. Then their summation corresponds to assuming that distributions over hypotheses generating voting elements are independent, i.e. $\forall i, j : p(x_i|I_i) \perp p(x_j|I_j)$. This **independence assumption** is clearly extremely crude. For instance, if voting elements i and j are adjacent in the image, then there is obviously a strong correlation between the hypothesis they come from, namely that they are very likely to be generated from the same object (or background clutter). Non-maxima suppression which is routinely performed within Hough transform can be regarded as a trick that compensates for the limitations of this independence assumption. As we will show later, the need for non-maxima suppression goes away if we do not make the assumption.

As an aside, the interpretation of the votes as log-likelyhoods, contradicts the way they are used within most of the approaches, as the votes are typically positive and span a limited range, and secondly, in learning-based frameworks the votes are learned as non-parametric probability densities and not log-likelihoods.

3.2 The probabilistic framework

Hough voting builds on the fact that pseudo-likelihoods (votes) $V(x_i = h|I_i)$ can be easily defined or learned from the data. However, rather than fusing all the votes in a principled way, Hough transform takes the easiest and fastest path and simply sums them.

As such summation has no probabilistic interpretation, our framework departs from Hough voting framework in the way, in which the votes $V(x_i = h|I_i)$ are fused. Rather than summing these votes, we model the joint distribution over all the random variables $\mathbf{x} = \{x_i\}$ and $\mathbf{y} = \{y_h\}$ in a probabilistic way, so that we can determine their values via the (MAP-) inference process. Thus, we are interested in modeling the joint posterior of $\mathbf{x}$ and $\mathbf{y}$ given image $\mathbf{I}$, where by image we mean the collection of the voting elements. Applying Bayes theorem then gives:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{I}) \propto p(\mathbf{I}|\mathbf{x}, \mathbf{y}) \cdot p(\mathbf{x}, \mathbf{y}) \quad (1)$$

Our model is illustrated in figure 2. We now focus on the likelihood and prior terms separately.

Likelihood Term. We make a different independence assumption to handle the likelihood term $p(\mathbf{I}|\mathbf{x}, \mathbf{y})$. We assume that given the existing objects $\mathbf{y}$ and the hypotheses assignments $\mathbf{x}$, the distributions of the appearances

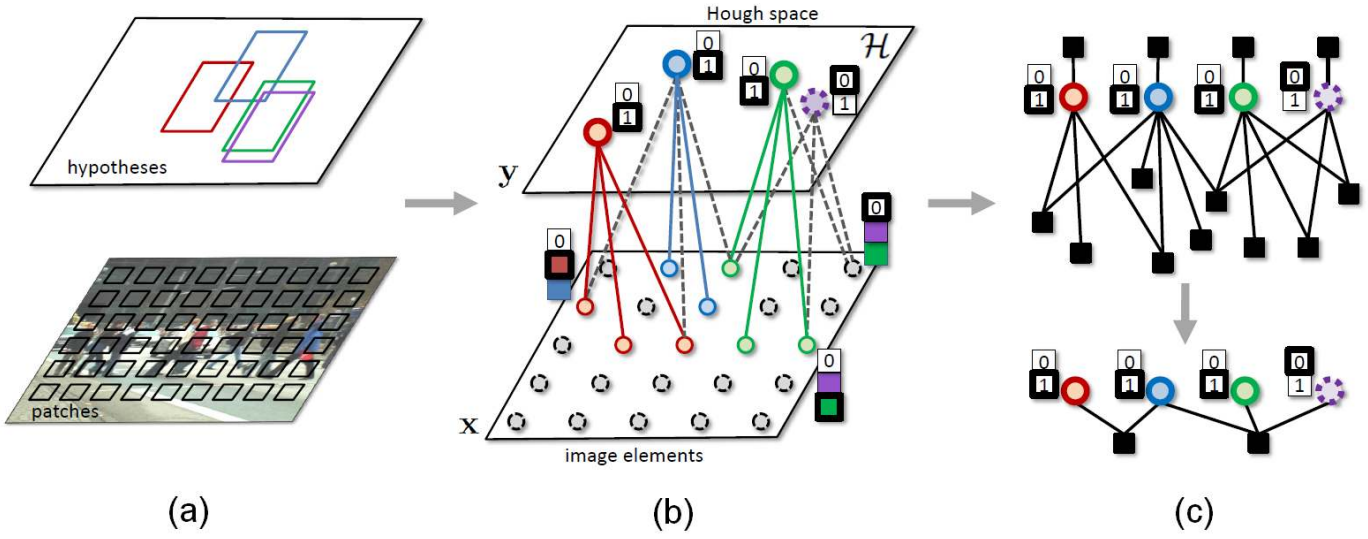


Fig. 2. Our probabilistic framework (please view in color). **(a)** The two key components of our framework: voting image-elements and the space of all possible object detections (Hypothesis space). The figure shows densely sampled image patches as the voting elements, and the bounding boxes corresponding to the 4 different object-detection hypothesis. **(b)** The figure shows the hypothesis variables $\mathbf{y}$ corresponding to the object-detection bounding boxes shown in (a), and their interaction with the voting variables $\mathbf{x}$. The lines indicate the support of detection hypotheses by voting elements. An example consistent labeling of variables is shown (the voting elements take the color of the hypothesis variable which explains them. Gray voting elements are explained by the background hypothesis). **(c)** shows the factor graphs representing the posterior distribution of the hypothesis variables (represented by circles) obtained from the joint posterior distribution p(x,y—I) (equation 8) by 'maximizing out' the voting element variables $\mathbf{x}$. The unary factor on the top of each hypothesis variable encodes the MDL prior (see equation 7). Other factors correspond to voting elements. The factor graph on top can be simplified by merging factors to give the factor graph at the bottom.

of voting elements are independent, i.e.:

$$p(\mathbf{I}|\mathbf{x},\mathbf{y}) = \prod_{i=1}^{N} p(I_i|\mathbf{x},\mathbf{y}). \tag{2}$$

Furthermore, we assume that the descriptor I_i of the voting element i depends only on the object assignment x_i, and is conditionally independent of the assignments of the remaining voting elements and the existence of all other objects in the image. Thus, we get:

$$p(\mathbf{I}|\mathbf{x},\mathbf{y}) = \prod_{i=1}^{N} p(I_i|x_i) \tag{3}$$

At a first glance, these assumptions may seem quite crude as the appearance of the element is assumed to be dependent only on the hypothesis x_i and conditionally independent from other voting elements and hypotheses. However, this dependence still may encode the relative positioning of the element i and the object corresponding to the hypothesis x_i. For instance, in the case of car detection, the expression $p(I_i|x_i)$ may model the appearance of the voting element (e.g. interest point) as a random variable dependent on the part of the car it comes from. We conclude the derivation of the likelihood part, by applying Bayes theorem once more and then omitting the terms that are constant for the given image:

$$p(\mathbf{I}|\mathbf{x},\mathbf{y}) = \prod_{i=1}^{N} p(I_i|x_i) \propto \prod_{i=1}^{N} \frac{p(x_i|I_i)}{p(x_i)}\,. \tag{4}$$

As a result, (1) can be rewritten as a following product:

$$p(\mathbf{x},\mathbf{y}|\mathbf{I}) \propto \prod_{i=1}^{N} p(x_i|I_i) \cdot \frac{p(\mathbf{x},\mathbf{y})}{\prod_{i=1}^{N} p(x_i)}. \tag{5}$$

Our expression for the likelihood is very similar to the (multiplicative) Hough transform as the data-dependent term turns out to be the product of terms $p(x_i|I_i)$, which are related to Hough votes. The terms are represented by the factors between different hypothesis variables in the factor graph shown in figure 2(c).

Prior terms. Before formulating the prior distribution $p(\mathbf{x},\mathbf{y})$, we should note that not all configurations $(\mathbf{x},\mathbf{y})$ are valid. If a voting element i is assigned to a non-background hypothesis h ($x_i = h$) then the hypothesis h must correspond to an existing object, i.e. y_h must be 1. Thus, the configuration is valid if and only if $y_{x_i} = 1, \forall x_i$. To avoid treating the background assignments $x_i = 0$ as a special case, we introduce the background hypothesis variable y_0, which is always set to 1. As a result the

consistency of the configuration $(\mathbf{x}, \mathbf{y})$ may be expressed by the hard constraint $\prod_{i=1}^{N} y_{x_i} = 1$.

We then assume that for all valid configurations our prior factorizes into products of priors on $\mathbf{y}$ and individual x_i resulting in the following expression:

$$p(\mathbf{x}, \mathbf{y}) = Z_1 \prod_{i=1}^{N} y_{x_i} \cdot p(\mathbf{y}) \cdot \prod_{i=1}^{N} p(x_i) \tag{6}$$

In this work we also focus on a very general prior on $\mathbf{y}$ (*Occam razor* or *MDL prior*) that simply penalizes the number of the active hypotheses $\sum_{h \in \mathcal{H}} y_h$, preferring explanations of the scene with as few objects as possible:

$$p(\mathbf{y}) = Z_2 \exp\left(-\lambda \sum_{h \in \mathcal{H}} y_h\right) = C_2 \prod_{h \in \mathcal{H}} \exp\left(-\lambda y_h\right) . \tag{7}$$

In (6-7), Z_1 and Z_2 are the normalization constants.

As a result of substituting (6) and (7) into (5), we get the final expression for the posterior:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{I}) \propto \prod_{i=1}^{N} p(x_i|I_i) \cdot \prod_{i=1}^{N} y_{x_i} \cdot \prod_{h \in \mathcal{H}} \exp\left(-\lambda y_h\right) \tag{8}$$

Note, that there might be several other approaches to choosing the prior distribution $p(\mathbf{x}, \mathbf{y})$. E.g. it may be computationally feasible to impose Potts prior on $\mathbf{x}$ ("if a voting element i is assigned to a hypothesis h, then the adjacent voting element j is also likely to be assigned to a hypothesis h"). The use of the Potts prior, however, is known to be detrimental for thin objects, e.g. lines [21].

It is also easy to introduce the standard non-maxima suppression via the overlap criterion into our framework. For this, one simply needs to define a prior that assigns zero probability to all configurations $\mathbf{y}$ where there exists a pair of enabled hypotheses with the bounding boxes overlapping too much. However, in our experiments, we refrain from using such a prior. This allows us to contrast our approach against traditional non-maxima suppression, and also to detect strongly overlapping object instances.

Algorithm 1 Greedy inference with dense hypothesis set

```
Detections = {∅};
for all elements i do
    x_i^cur = 0; // "explaining" all elements by background
end for
// iterated Hough voting:
loop
    // initializing Hough map to zero:
    for all hypotheses h do
        M(h) = 0;
    end for
    // casting Hough votes:
    for all elements i do
        for all hypotheses h "near" element i do
            M(h) += max(log P(x_i=h|I_i) −
                        log P(x_i=x_i^cur|I_i), 0);
        end for
    end for
    // locating the hypothesis with the highest score:
    y = arg max(M);
    if M(y) ≤ λ then
        return Detections; // detection set is greedy-optimal
    end if
    Detections.Add(y);
    for all elements i "near" hypothesis y do
        if P(x_i=y|I_i) > P(x_i=x_i^cur|I_i) then
            // updating the "explanation" for this i:
            x_i^cur = y;
        end if
    end for
end loop
```

Fig. 3. The pseudocode for the greedy inference with dense hypothesis set. The informal predicate *element i is "near" hypothesis y* should be interpreted as follows: an element i might vote for a hypothesis y for some image appearance (such as e.g. an edge pixel i that is located near a straight line h, or an image patch i that is located inside a detection box h).

4 Inference

Log-posterior maximization. In the paper, we focus on computing the maximum-a-posteriori (MAP) configurations (MAP-inference) under the probability model (8). By taking the logarithm of (8), the MAP-inference in our model is rendered as the maximization problem for the following log-posterior function:

$$E(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} u_i(x_i) + \sum_{h \in \mathcal{H}} v_h(y_h) + \sum_{h \in \mathcal{H}} \sum_{i=1}^{N} w_{ih}(x_i, y_h) , \tag{9}$$

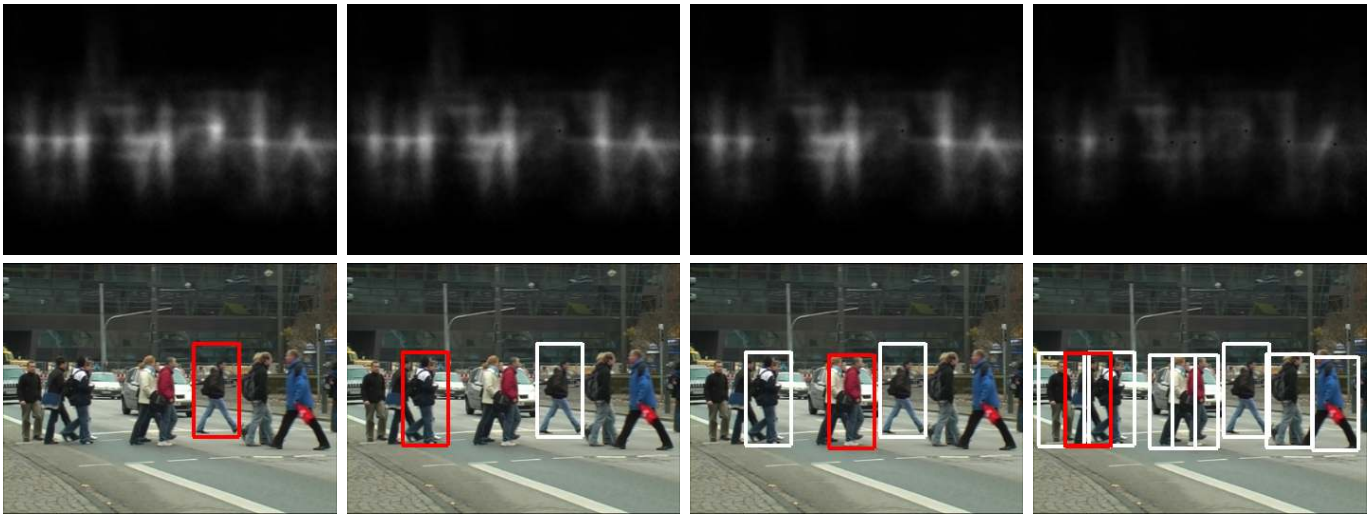
where $u_i(x_i) = \log p(x_i|I_i)$, $v_h(y_h) = -\lambda y_h$, and $w_{ih}(x_i, y_h) = -\infty$ if $x_i = h$ and $y_h = 0$ and $w_{ih}(x_i, y_h) = 0$ otherwise.

The graphical model interpretation of the energy (9) (Figure 2 (b)) reveals that the associated graph has a "bipartite"structure. Thus, one can choose the optimal value of x_i independently, if the configuration of $\mathbf{y}$ is given. Thus, the log-posterior (9) can be represented only as the function of the hypotheses variables $\mathbf{y}$, after the $\mathbf{x}$ variables are "maximized out":

$$E_y(\mathbf{y}) = \max_{\mathbf{x}} E(\mathbf{x}, \mathbf{y}) = \sum_{h \in \mathcal{H}} -\lambda y_h + \sum_{i=1}^{N} \max\left(\max_{h: y_h = 1} u_i(h), u_i(0)\right) \tag{10}$$

This simplified factor graph is shown in the bottom half of figure 2 (c).

Sparse set of hypotheses. We tried many different algorithms for performing MAP-inference in our model



1st iteration 2nd iteration 3rd iteration 8th iteration

Fig. 4. Greedy MAP-inference in our model for pedestrian-detection example from Figure 1. For each iteration, we give the Hough image M^t (top) and highlight in red the detection corresponding to its maximum (bottom). Note how the Hough images $M^t(h)$ are changed between iterations, so that implicit "non-maximum suppression" driven by the probability function is performed. As a result, multiple pedestrians are detected despite significant overlaps between them.

including performing loopy belief-propagation [22] in the bipartite graph defined by (9). The special form of the pairwise terms permits a very compact message representation (the same as used in the affinity propagation [23]). We have also tried simulated annealing optimization for the binary-labelled function (10).

Both loopy belief propagation (LBP) and simulated annealing (SA) were not able to handle the very high order potentials present in our model. To overcome this problem, we adaptively reduced the size of our hypothesis space. We did this by using standard Hough voting to find (sample) a moderately large number (dozens to hundreds) of peaks in the Hough image. We then restrict the Hough space $\mathcal{H}$ to these peaks. As the majority of voting element vote for a limited number hypotheses each ($p(x_i|I_i) = 0$ for many assignments values of x_i), we were able to reduce the size of the Hough space considerably without loss of many energy terms.

In our experiments LBP and SA gave reasonable results with the adaptive sparsification heuristics discussed above. However, they were still quite computationally expensive. Also the inability of these inference methods to handle large set of hypotheses is a significant limitation which potentially can lead to loss of detections and lower recall rate of object detection performance.

Submodularity and connection with uncapacitated facility location problem.

The maximization of (10) can be viewed as the well studied in operation research community *facility location* task, that considers the problem of optimal placement of facilities (detected objects) in order to minimize transportation costs (negative votes from voting element). One of the well-known properties of the objective function of facility location problem (10) is it's *submodularity* (see e.g. [24]) Unlike the problem of minimizing submodular functions, the problem of maximizing submodular functions is NP-hard. But approximations have been studied extensively for both the general task of submodular function maximization and the particular problem of facility location. The best approximation factor known for facility location is 0.828 that is achieved by polynomial-time algorithm based on the idea of randomized rounding [25].

The greedy algorithm, that iteratively augments a current solution with an element of maximum incremental value, is proven to have an approximation factor 0.632 for the task of submodular functions maximization [26]. This simple method has been shown to be an efficient heuristic for both maximizing submodular functions over different constraint structures (e.g. [27]) and facility location problem (e.g. [28]). Not surprisingly, in our framework greedy algorithm showed approximately the same accuracy as LBP and SA. Moreover in contrast to LBP and SA, it turned out that the iterative greedy inference doesn't require reducing the hypothesis space. This property potentially allows greedy algorithm to achieve higher recall compared to LBP and SA.

Iterative sampling with dense set of hypotheses. The greedy iterative algorithm starts with all y_h set to 0 and x_i set to 0 (background). In step t the algorithm makes a hypothesis h^t active (by setting $y_{h^t} = 1$), simultaneously switching some of x_i to h^t (x_i is switched to h^t only if this increases the posterior). The hypothesis h^t is picked so that the biggest increase of the posterior is obtained.

In each iteration, it identifies the optimal hypothesis h^t to be made active by using Hough voting. In iteration

t, a voting element i casts an (additive) vote:

$$V_i^t(h) = \max\left(\log P(x_i{=}h|I_i) - \log P(x_i{=}x_i^t|I_i), 0\right) , \quad (11)$$

where x_i^t denotes the hypothesis that the voting element i is assigned by the step t. Each vote thus encodes the potential increase of the log-posterior part for the element i, should the hypothesis h be enabled.

The votes are accumulated in a Hough image $M^t(h) = \sum_{i=1}^{N} V_i^t(h)$. Then, the maximal value hypothesis $h^t = \operatorname{argmax}(M^t(h))$ is considered. If $M^t(h^t)$ is less or equal λ then the algorithm terminates, as the log-posterior (10) cannot be increased any further in a greedy way. Otherwise, y_{h^t} is set to 1 and each x_i is switched to h^t (i.e. x_i^{t+1} is set to h^t), provided that this increases the log-posterior. The pseudo-code for the full algorithm is provided in figure 3.

Several optimizations can be performed to ensure a reasonable computational cost. The votes in each iteration $\log P(x_i = y_{x_i^t}|I_i)$ can be stored from the previous iteration rather than recomputed afresh each time. More generally, the new Hough image M^t can be in many cases computed incrementally from M^{t+1} by subtracting the previous votes and adding the new votes for the voting elements i that have changed x_i.

The algorithm presented above is only greedy in its selection of the active hypothesis. It handles the voting element variables in a principled fashion and allows them to switch their votes. In this sense, it is better than the traditional heuristic method for multiple hypothesis selection which involves iteratively selecting a hypothesis and then deleting the votes of all elements that voted for it. This strategy is greedy in the labeling of both hypothesis variables and voting element variables. Once a hypothesis is chosen, the votes cast by elements voting for this hypothesis are fixed and cannot be changed.

5 Experiments

In this section, we present the experimental evaluation of the proposed approach. Thus, we consider three applications: the edge-based straight line detection (the classical application of Hough transform), the part-based object class (pedestrian) detection in images (continuing the line of works started by Implicit Shape Models), and the cell counting task in microscopic images.

In the proposed comparisons, we considered the following baseline approaches:

1) **Hough transform followed by non-maximum suppression.** For this baseline, we take a traditional approach of computing the Hough map, and then greedily picking a set of peaks, starting with the most prominent one. Every time a peak is selected, the Hough map within the radius R from the selected peak is suppressed (set to zero). The process ends once the maximal value in the map falls below the threshold τ.
2) **Nullifying votes of "explained" elements.** This baseline closely follows the proposed greedy approach, except for one thing: once a hypothesis corresponding to a maximum is selected at each iteration, all the elements that vote for this hypothesis with positive vote (11) stop voting for all other hypotheses on further iterations (their votes are set to zero). This corresponds to nullifying the votes of the elements once they get explained by an activated hypothesis. Such heuristics can also be regarded as the "double-greedy" optimization of the energy (9), when both y-variables and x-variables are optimized in a greedy fashion. Thus, x variables in this case are never changed, once they switch from the background to some other label (unlike the proposed more powerful "single-greedy" algorithm, where the change in y-variables takes into account and leads to the optimal change of x-variables).
3) **Mode-seeking**. While the two baselines described above provided reasonably competitive performance in the experiments, we also consedered the use of mode-seeking, which is popular alternative to non-maxima suppression within Hough transform [3]. Thus, in the line detection experiment, we also tried the medoid-shift algorithm [29] to prune the set of local maxima in the Hough map. To make a comparison more favourable for this baseline, the value of the bandpass parameter σ in the medoid-shift algorithm was chosen by optimization on the test set. We thus run the medoid-shift algorithm leaving only the maxima that were found to be the medoids. Unfortunately, the results we got using the medoid-shift algorithm were significantly and uniformly worse than those with non-maxima suppression (the weaker of the remaining baselines), therefore we do not report them in a sequel.

5.1 Line detection

Experimental protocol. We first start with the classical problem of line detection in images. As a benchmark, we considered the YorkUrbanDB dataset [30]. The dataset contains 102 images of urban scenes, of which 20 were used for parameter validation and 82 for testing the performance. The scenes in the dataset have predominantly "Manhattan" geometry, with the majority of straight lines belonging to the three orthogonal families. The authors of the dataset also provide a set of "Manhattan" line segments semi-automatically annotated in each image as well as the locations of "Manhattan" vanishing points.

Given a set of straight lines detected with some algorithm, we define the *recall* to be the ratio of the straight segments that lie within 2 pixels from one of the detected lines (a generous 2 pixel threshold was allowed to account both for discretization effects and

. available at `http://www.elderlab.yorku.ca/YorkUrbanDB/`

Fig. 5. Sample detections for the proposed framework (top) and Hough transform+NMS (bottom). The λ and τ parameters were set so that both methods detect on average 50 lines per image. Note, how the proposed framework is able to discern very close yet distinct lines, and is in general much less plagued by spurious detections.

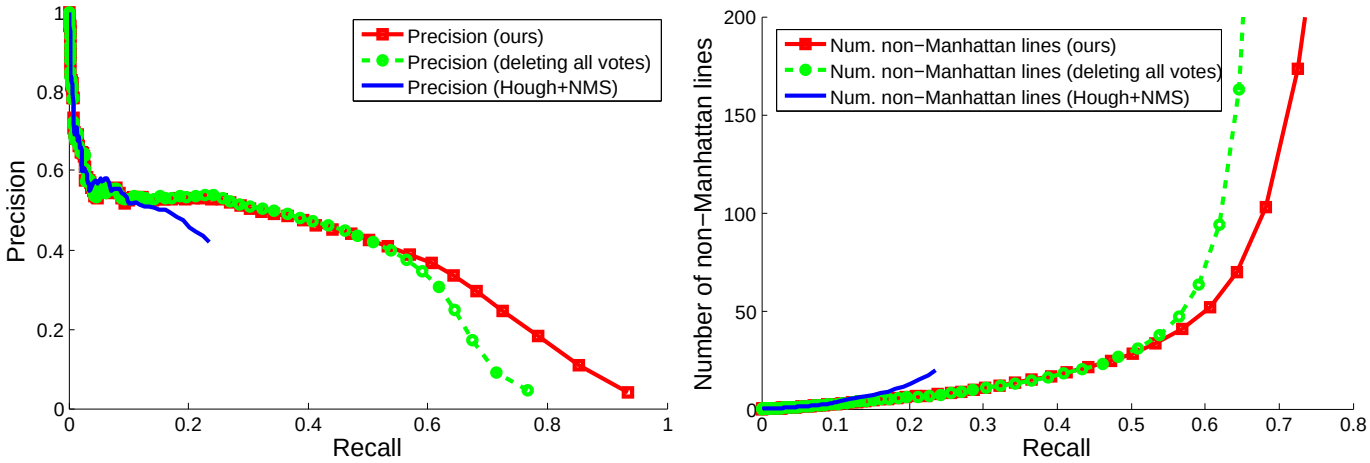


Fig. 6. Recall-precision and recall-(average number of non-Manhattan lines) curves for the proposed framework and for the two baseline approaches. For a given recall, the proposed method find bigger share of lines that pass near annotated segments (left). The tends to be fewer "non-Manhattan" lines in the output of the proposed method (right), which, for the York Urban dataset, indirectly implies smaller number of spurious detections. See the text for more discussion on the experimental protocol.

for the edge detector imperfections). We also considered two measures of *precision*. For the first, more traditional measure, we just matched detected lines to the ground truth segments that lied within 2 pixels from them (at most one line could be matched to each segment), and counted the ratio of matched lines to all lines. This measure however often penalizes correctly detected lines that were not annotated. We therefore computed the second measure of precision by counting the number of non-Manhattan lines treating them as errors. Such approach still penalizes correctly detected non-Manhattan lines, but there are few of them in the particular dataset, we have considered. To determine whether a line is a Manhattan one, we look at the angles between the line and the directions towards the ground truth vanishing points from the midpoint of the part of the line visible in the image. If all three angles are greater than 2 degrees, then we treat the line as non-Manhattan and erroneous detection.

Algorithmic details. For each image a Canny edge detection (OpenCV implementation, Canny thresholds = 700 and 1400) was performed. Each edge pixel was considered a voting element, described by its position I_i and its local orientation n_i in the image plain. We defined:

$$p(x_i = l|I_i) = Z_3 \exp\left(-C_1 \operatorname{dist}(i,l) - C_2 \operatorname{angle}(i,l)\right) \quad (12)$$

$$p(x_i = 0|I_i) = Z_3 \exp\left(-C_0\right) , \quad (13)$$

where $\operatorname{dist}(i,l)$ is the distance between the edge pixel i and the line l, $\operatorname{angle}(i,l)$ is an angle between the direction

of image gradient in the edge pixel i and the line l, Z_3 is a normalizing constant, C_0, C_1, C_2 are constants set up by validation. We then used the greedy version of the proposed framework to detect the lines in the images.

As discussed above, for benchmarking, we compared the results of the proposed framework with the results of the Hough transform followed by non-maximum supression and with the version with nullifying the votes. We used the "soft" voting scheme, where each edge pixel i voted for the line l with the strength $\max(C_0 - C_1 \operatorname{dist}(i,l) - C_2 \operatorname{angle}(i,l), 0)$, so that the Hough images produced during Hough voting were essentially the same as on the first step of the proposed greedy algorithm. We then identified the local maxima in the Hough images, and performed non-maxima suppression. These required some reasonable distance measure between the lines, for which we used the following. Given two lines l_1 and l_2, we again clipped them against the image boundaries obtaining segments s_1 and s_2. We then defined the distance between lines l_1 and l_2 to be the maximum over 4 numbers corresponding to the distances from each endpoint of each clipped segment (s_1 and s_2) to the other line (l_2 or l_1 respectively). The minimal distance R within non-maximum suppression as well as the optimal C_2 were set up by validation.

Results. Quantitative comparisons between the proposed method and the first two baselines baselines were summarized via Recall-Precision curves generated by varying λ parameter in the case of the proposed method as well as the 2nd baseline and the τ threshold in the case of the Hough transform + non-maximum suppression (the 1st baseline). The curves for the test set are given in Figure 6. As can be seen, the proposed approach outperforms both baselines considerably with respect to both precision measures. In particular, the optimal maximum suppression radius for the baseline algorithm ($R = 13$) makes the first baseline algorithm unsuitable when higher values of recall are desired.

Some qualitative examples at a low recall-high precision value are given in Figure 5. We note, that the second baseline is much more competitive but still performs worse than the proposed approach, particularly at high recall rate. This is because achieving high recall requires the detection of spatially close lines that "compete" for voting elements with each other. In this regime, nullifying the residual votes rather than keeping them is most detrimental.

5.2 Pedestrian detection

Experimental protocol. We now describe the experiments on detection of pedestrians in street images. We downloaded two video sequences *TUD-campus* and *TUD-crossing* containing mostly profile views of pedestrians in relatively crowded locations. The original annotations provided with [31] included only the pedestrians occluded by no more than 50%. As we were interested in the performance of the method under significant overlap, we re-annotated the data by marking all pedestrians whose head and at least one leg were clearly visible. After reannotation, the *TUD-campus* and *TUD-crossing* sequences contain 71 images with 304 ground truth bounding boxes annotated, and 201 images with 1018 bounding boxes accordingly.

To obtain the probabilistic votes we used the Hough forest [5] learned on the separate training dataset (considered in [5]). Hough forests are learned on a dataset of 16x16 patches extracted from images with the objects of interest (pedestrians) at a fixed scale, and from the set of background images. After training, Hough forests are able to map the patch appearance and location (encoded by I_i) directly to $p(x_i|I_i)$, which is exactly what is needed in the proposed framework.

Algorithmic details. For single-scale scenario, Hough forests can be seamlessly incorporated into the proposed framework. Full version of the greedy algorithm is directly applicable here, with both set of the voting elements and set of possible centroid locations being the set of all pixels (Figure 4 provides an example of the greedy algorithm in a single-scale case). We were however interested in the detection in a multiscale setting, in which the Hough space is parameterized by the centroid location χ_i and the scale s_i of the pedestrian. To get an estimate of $p(x_i = (\chi_i, s_i)|I_i)$ we apply Hough forest to the image resized by the scale s_i.

In our experiments the set of voting elements was parameterized by pixels at the largest scale. So the number of voting elements at different scales is constant, but as objects at different scales are of different sizes, detection of larger objects should require more evidence than detection of a smaller object. This was achieved by increasing penalty for detection of larger objects. So in the experiments we upscaled λ proportionally to the area of objects at a particular image scale. The background probability $p(x_i = 0|I_i)$ in the proposed framework was set to a $p_0 \max_s p(x_i = 0|I_i, s)$, where p_0 is a constant chosen on the validation set and $p(x_i = 0|I_i, s)$ is the output of Hough forests applied to the i-th patch at s-th scale. Naturally, when a detection at a particular scale is selected by the greedy algorithm, this leads to the updates of the votes (11) of the adjacent pixels at all scales (as the scale is considered simply as a part of the configuration space in the Hough voting).

. While it is possible to introduce an additional variance parameter σ into the exponent in (12), a careful inspection reveals that this would not change the family of energies (9) spanned by different C_0, C_1, C_2 and λ.

. tuned on the validation set to maximize the area under recall-precision curve for high precision values.

. In more detail, we considered an excessive number of 6000 local minima in each Hough image with the highest values. We then truncated the curve at the point corresponding to $\tau = \max_{\text{test set}} M_{6000}$, where M_{6000} denotes the value corresponding to the 6000th strongest local maximum in the Hough image. This point corresponded to approximately 0.25 recall. Note that changing R would not allow to extend the curve into higher recall region without the drammatic increase of the "6000 local minima per image" budget.

. available at `http://www.mis.tu-darmstadt.de/node/382`

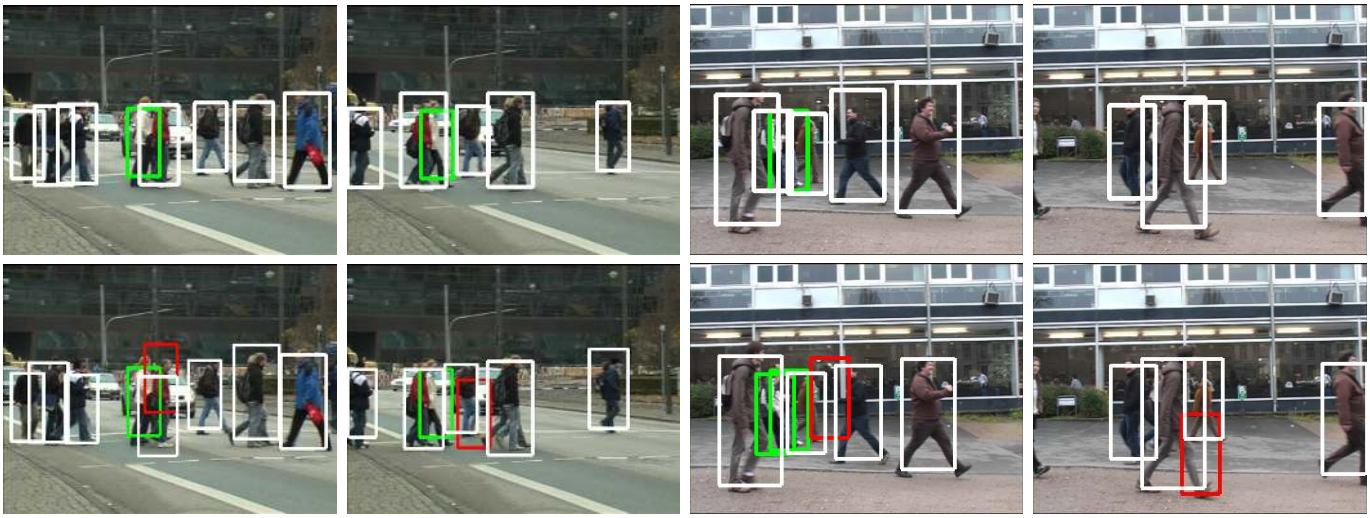


Fig. 7. Sample detection results for the proposed framework (top) and Hough transform+NMS (bottom) for the TUD-crossing and TUD-campus sequences at equal error rates (white = correct detection, red = false positive, green = missed detection). Note how the proposed framework is capable of detecting strongly overlapping objects without producing many false positives.

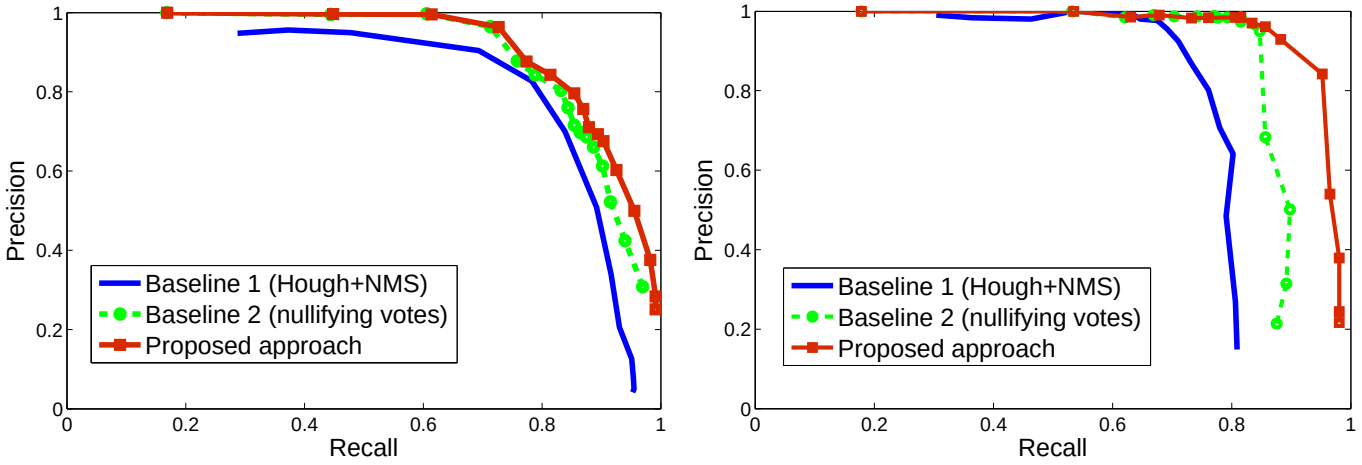


Fig. 8. Precision-recall curves for the proposed framework (red), nullifying votes of "explained" elements(green), and Hough transform+NMS (blue) on *TUD-crossing* (left) and *TUD-campus* (right) sequences. The proposed approach achieves better performance on these datasets containing a significant number of object overlaps.

Thus we merge together the voting elements at different scales into a single voting element that has to choose a hypothesis to vote for (a slight subtlety here is that the descriptor I_i is different for different scales. We note that more rigorous way to perform multi-scale object detection would involve using explicit multi-scale version of Hough forests regression).

The performance of the proposed framework was compared with both baselines, of which the 1st one is extremely close to [5] and uses non-maxima suppression based on the overlap criterion. The overlap threshold within the NMS was set up by cross-validation. For both algorithms, we used one of the sequences for the validation and then tested on the other.

We used 3 scales for the *TUD-crossing* sequence and 5 scales for the *TUD-campus* sequence all differing by a factor of 0.85. For the matching between the produced set of detections and the ground truth, the standard 50% overlap criterion was used.

Results. Resulting recall-precision curves (generated in the same way as in the line detection experiments) are shown in Figure 8. Hough voting [5] with non-maxima suppression fails to achieve high recall values in multi-scale setting. This happens in part because close peaks corresponding to the same object arise in Hough images of different scales, and non-maximum suppression could not filter out these duplicated detections without filtering out close correct detections as well (see Figure 7). The proposed framework does not require discerning

. the minor difference is that we do not interpolate linearly between scales in order to obtain the exact scale of the detection, while [5] does. This affects all algorithms under comparison equally.

between such peaks and thus shows better performance on both datasets. As in the case with line detection and largely for the same reasons, the second baseline performs better than Hough transform+non maxima suppression but still worse than the proposed approach, particularly at higher recall rates.

5.3 Cell detection and counting

Experimental protocol. Finally, we considered a rather different modality of fluorescent microscopy and the task of cell counting. We followed the experimental setup of [32], and considered the dataset introduced in [32] and generated using the tool [33] (an example is in Figure 5.3a). The task is to estimate the number of cells in each image (the true number in each image varies and is on average equal to 171 ± 64 cells per image). The cells are often clamped together thus making the task of discerning adjacent detections a particularly difficult one.

We followed the splits and the experimental protocol suggested in [32], where the number of training images varies from 1 to 32 and in each case the same number of images is left for validating the meta-parameters. Also following [32], we considered the two ways to validate the detection algorithms in this case. In the first case, we tune the parameters to minimize the average absolute counting error on the validation set ('counting' validation). In the second case, we tune the parameters to minimize the detection errors. In more detail, the images are 'dotted' (one dot for each image). The produced detections were then matched to the ground truth using the Hungarian algorithm; the detection was allowed to match to the ground truth dot if they are within 8 pixels.

Algorithmic details. We have used the code of [5] to train a single-scale Hough forest for the dataset, without any modification to features used by [5]. During training, we assigned the pixels that were further than 10 pixels from the ground truth centers to the negative class, otherwise the pixel was considered a part of the positive detection with the closest center.

Results. Following [32], we report the counting accuracy (mean absolute counting error over 100 test images) obtained with the proposed framework as well as by the Hough voting+NMS baseline for the varying number of images in the training/validation sets (Table 1). For both methods we consider two different ways to validate meta-parameters described above. As can be seen, the proposed framework once again demonstrates lower mean error than the Hough+NMS baseline (for 16 and 32 training images the difference is statistically significant). We give the qualitative example of relative performance in Figure 5.3(b)-(c).

For the reference, we also reproduce the performance of the counting-by-detection baseline from [32]. There, the detection was obtained with a rather different approach (sliding window with linear SVM over the densely computed SIFT followed by non-maximal suppression). We also note that the counting framework suggested in [32] achieves uniformly lower counting error than all methods listed in Table 1. However, unlike the methods listed in the table, the counting framework [32] does not explicitly provide the set of detections. Depending on the ultimate application, the lack of the explicit detection list may or may not be acceptable.

6 Discussion

We have presented a framework for detecting multiple object instances in images, which is similar to the traditional Hough transform. It was demonstrated that by redeveloping Hough transform within a probabilistic, energy-based framework, one can avoid solving the tricky problem of distinguishing multiple local peaks in the Hough image. Instead, the greedy inference in our framework only requires picking the overall (global) maxima of a sequence of Hough images. In this way, non-maxima suppression step can be bypassed altogether, and, according to our experiments, a significant increase in accuracy can be obtained.

We believe that the suggested framework and the inference algorithms within it lend themselves to easy integration with other sources of information. In this way one may, for example, explicitly model the positioning of the vanishing points in the image and/or the horizon and the camera positioning. By connecting the respective variables with our $\mathbf{x}$ and $\mathbf{y}$ variables in a unified graphical model, one can perform joint inference over scene elements on multiple levels. This is investigated in our follow up work [34], where we consider a multi-layer graphical model for the parsing of an image of a man-made environment. The model [34] includes layers for a) estimation of straight lines based on edge pixels (this layer corresponds to the model derived and evaluated in this paper), b) grouping of the detected lines in parallel line families, c) the estimation of the horizon and the zenith of the scene.

The code for line and pedestrian detection based on greedy inference within our framework as well as the additional annotations for the TUD datasets publicly are publically available at the project webpage.

References

[1] P. Hough, "Machine analysis of bubble chamber pictures," in *Int. Conf. High Energy Accelerators and Instrumentation*, 1959.

[2] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

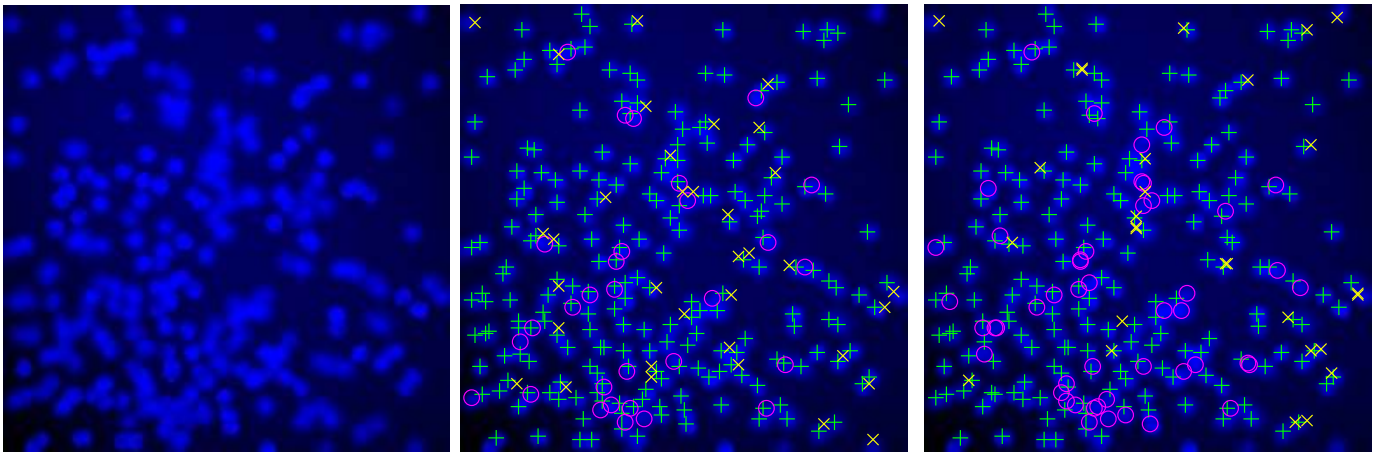
[3] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *IJCV*, vol. 77, no. 1-3, pp. 259–289, 2008.

[4] S. Maji and J. Malik, "Object detection using a max-margin hough tranform," in *CVPR*, 2009.

[5] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *CVPR*, 2009.

[6] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *CVPR*, 2009.

. http://graphics.cs.msu.ru/en/science/research/machinelearning/hough



(a) Input image (b) Detection (proposed) (c) Detection (Hough+NMS)

Fig. 9. Cell detection example. (a) – the input fluorescence image, where individual cells need to be detected. (b) and (c) – the result of the proposed framework and of the Hough+NMS baseline respectively. The two systems were trained on the same set of 32 images. Each $+$ denotes a correct detection (within 8 pixels from the ground truth dot), $\times$ denotes a false positive, $\bigcirc$ denotes a cell missed by the detector.

TABLE 1

Mean absolute errors for cell counting on the test set of 100 fluorescent microscopy images. The second column corresponds to the error measure used for learning meta-parameters on the validation set. The last 6 columns correspond to the numbers of images in the training and validation sets. The average number of cells is 171 ± 64 per image. Standard deviations in the table correspond to 5 different draws of training and validation image sets. The rows correspond to the proposed framework (top two lines), the Hough+NMS baseline (row 3 and 4). The bottom lines contain the results for the detection mathod evaluated in [32]. The counting framework in [32] achieves lower mean errors, but does not produce a set of detections. Please see text for more details.

	Validation	$N=1$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$
Proposed framework	counting	–	14.6 ± 5.6	11.4 ± 2.0	$\mathbf{9.3\pm2.8}$	7.9 ± 1.4	6.2 ± 0.2
Proposed framework	matching	$\mathbf{12.5\pm2.8}$	$\mathbf{11.4\pm2.6}$	$\mathbf{10.8\pm1.4}$	10.7 ± 2.8	$\mathbf{7.5\pm0.5}$	$\mathbf{6.0\pm0.5}$
Hough + NMS	counting	–	15.3 ± 3.6	16.5 ± 8.8	9.8 ± 1.9	9.0 ± 0.3	7.7 ± 0.4
Hough + NMS	matching	13.0 ± 3.1	17.2 ± 8.0	11.2 ± 1.5	10.7 ± 0.8	9.6 ± 1.4	8.1 ± 0.4
Sliding window + NMS (from [32])	counting	28.0 ± 20.6	20.8 ± 5.8	13.6 ± 1.5	10.2 ± 1.9	10.4 ± 1.2	8.5 ± 0.5
Sliding window + NMS (from [32])	matching	20.8 ± 3.8	20.1 ± 5.5	15.7 ± 2.0	15.0 ± 4.1	11.8 ± 3.1	12.0 ± 0.8

[7] R. Okada, "Discriminative generalized hough transform for object detection," in *ICCV*, 2009.

[8] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *ICCV*, 2009.

[9] R. S. Stephens, "Probabilistic approach to the Hough transform," *Image and Vision Computing*, vol. 9(1), pp. 66–71, 1991.

[10] T. Minka, "The 'summation hack' as an outlier model," *Microsoft Research Technical Report*, 2003.

[11] M. Allan and C. K. I. Williams, "Object localisation using the generative template of features," *Computer Vision and Image Understanding*, vol. 113, no. 7, pp. 824–838, 2009.

[12] D. Hoiem, C. Rother, and J. M. Winn, "3d layoutcrf for multi-view object class recognition and segmentation," in *CVPR*, 2007.

[13] N. Lazic, I. Givoni, B. Frey, and P. Aarabi, "Floss: Facility location for subspace segmentation," in *ICCV*, 2009.

[14] A. Delong, A. Osokin, H. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," in *CVPR*, 2010.

[15] L. Ladicky, C. Russell, P. Kohli, and P. Torr, "Graph cut based inference with co-occurrence statistics," in *ECCV*, 2010.

[16] A. Lehmann, B. Leibe, and L. V. Gool, "PRISM: PRincipled Implicit Shape Model," in *BMVC*, 2010.

[17] C. F. C. Desai, D. Ramanan, "Discriminative models for multi-class object layout," in *ICCV*, 2009.

[18] Y. G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning," *IJCV*, vol. 3, no. 1, pp. 73–102, 1989.

[19] S. C. Zhu and A. L. Yuille, "Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation," *TPAMI*, vol. 18, no. 9, pp. 884–900, 1996.

[20] Y. Amit, D. Geman, and X. Fan, "A coarse-to-fine strategy for multiclass shape detection," *TPAMI*, vol. 26, no. 12, pp. 1606–1621, 2004.

[21] V. Kolmogorov and Y. Boykov, "What metrics can be approximated by geo-cuts, or global optimization of length/area and flux," in *ICCV*, 2005, pp. 564–571.

[22] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Palo Alto: Morgan Kaufmann, 1988.

[23] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.

[24] U. Feige, V. S. Mirrokni, and J. Vondrak, "Maximizing non-monotone submodular functions," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 461–471.

[25] A. A. Ageev and M. I. Sviridenko, "An 0.828-approximation algorithm for the uncapacitated facility location problem," *Discrete Applied Mathematics*, no. 93, pp. 289–296, 1999.

[26] G. Nemhauser, L. Wolsey, and M. L. Fisher, "An analysis of the approximations for maximizing submodular set functions - 1," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[27] M. Conforti and G. Cornuejols, "Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem," *Discrete Applied Mathematics*, pp. 251–274, 1984.

[28] D. Hochbaum, "Heuristics for the fixed cost median problem," *Math. Programming*, pp. 148–162, 1982.
[29] Y. Sheikh, E. A. Khan, and T. Kanade, "Mode-seeking by medoidshifts," in *ICCV*, 2007.
[30] P. Denis, J. H. Elder, and F. J. Estrada, "Efficient edge-based methods for estimating manhattan frames in urban imagery," in *ECCV*, 2008.
[31] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking." in *CVPR*, 2008.
[32] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *NIPS*, 2010.
[33] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja, "Computational framework for simulating fluorescence microscope images with cell populations," *TPAMI*, vol. 26, no. 7, pp. 1010–1016, 2007.
[34] O. Barinova, V. Lempitsky, E. Tretiak, and P. Kohli, "Geometric image parsing in man-made environments," in *ECCV*, 2010.

Olga Barinova Olga Barinova has received her PhD in 2010 from Lomonosov Moscow State University. During her PhD studies she has done an internship in Microsoft Research Cambridge. Currently she is a researcher at Lomonosov Moscow State University. Olga's research interests include various aspects of machine learning and computer vision.

Victor Lempitsky Victor Lempitsky received his PhD degree from Moscow State University in 2007. During his PhD studies he also was an intern with the University of Western Ontario and with Microsoft Research Cambridge. He then hold postdoctoral positions with the Computer Vision group, Microsoft Research Cambridge and with the Visual Geometry group, University of Oxford. Currently, Victor is a researcher with Yandex, Moscow. Victor's research interests are in various aspects of computer vision and biomedical image processing.

Pushmeet Kohli Pushmeet Kohli is a research scientist in the Machine Learning and Perception group at Microsoft Research Cambridge, and an associate of the Psychometric Centre and Trinity Hall, University of Cambridge. His research revolves around Intelligent Systems and Computational Sciences, and he publishes in the fields of Machine Learning, Computer Vision, Information Retrieval, and Game Theory. His current research interests include human behaviour analysis and the prediction of user preferences.

Pushmeet has won a number of awards and prizes for his research. His PhD thesis, titled "Minimizing Dynamic and Higher Order Energy Functions using Graph Cuts", was the winner of the British Machine Vision Associations Sullivan Doctoral Thesis Award, and was a runner-up for the British Computer Society's Distinguished Dissertation Award. Pushmeets was also one of the two United Kingdom nominees for the ERCIM Cor-Bayern award in 2010. Pushmeets papers have appeared in SIGGRAPH, NIPS, ICCV, AAAI, CVPR, PAMI, IJCV, CVIU, ICML, AISTATS, AAMAS, UAI, ECCV, and ICVGIP and have won best paper awards in ICVGIP 2006, 2010 and ECCV 2010. His research has also been the subject of a number of articles in popular media outlets such as Forbes, The Economic Times, New Scientist and MIT Technology Review.