Detecting Partially Occluded Objects with an Implicit Shape Model Random Field

Paul Wohlhart, Michael Donoser, Peter M. Roth, and Horst Bischof

Graz University of Technology, Institute for Computer Graphics and Vision
{wohlhart,donoser,pmroth,bischof}@icg.tugraz.at

Abstract. In this paper, we introduce a formulation for the task of detecting objects based on the information gathered from a standard Implicit Shape Model (ISM). We describe a probabilistic approach in a general random field setting, which enables to effectively detect object instances and additionally identifies all local patches contributing to the different instances. We propose a sparse graph structure and define a semantic label space, specifically tuned to the task of localizing objects. The design of the graph structure then allows to define a novel inference process that efficiently returns a good local minimum of our energy minimization problem. A key benefit of our method is, that we do not have to fix a range for local neighborhood suppression, as necessary for instance in related non maximum suppression approaches. Our inference process implicitly is capable to separate even strongly overlapping object instances. Experimental evaluation compares our method to state-of-the-art in this field on challenging sequences showing competitive and improved results.

1 Introduction

Localizing instances of arbitrary categories in cluttered scenes is one of the main challenges in computer vision. In general, most methods learn appearance and spatial relation models of the categories from labeled training images and use the obtained models to localize previously unseen instances in test images.

Currently, mainly two different approaches can be distinguished: (a) sliding window based and (b) part based methods. Sliding window based methods like [1, 2] evaluate powerful classifiers on windows at all possible image locations, analyzing discriminative local descriptors like the histogram-of-gradients [3]. Although, these approaches have shown to provide excellent results for rectangular shaped categories, they yield limited performance for deformable objects. Thus, recently part based models have become more popular. The notion of parts has a long history in computer vision, starting from the Pictorial Structures model [4], where each object is represented as an assembly of local parts and flexible spatial relations between them. While early work in this field manually identified semantically meaningful parts, recent research [5, 6] focused on how to automatically select discriminative parts from training data.

Part-based models mainly differ in the way the spatial relations between the individual parts are defined, ranging from fully connected models, where each part is connected to all other parts (constellation model [7]), to models without any spatial relation (bag-of-words model [8]). Thus, they mainly differ concerning their inference complexity, where a constellation model for example is only able to handle a few parts and additionally has to significantly reduce the number of part candidates in each test image.

Recently, tree models have become the most popular spatial model for part-based recognition due to the highly efficient inference possibilities for tree structures. Such tree models for example have led to the deformable part model [6], one of the most successful algorithms on the PASCAL Visual Object Class (VOC) challenge.

One of the first tree-shaped models for the task of object detection focussed on a specific sub-type, the star shaped model, where each part is only connected to a centroid part. The underlying representation was denoted as the *Implicit Shape Model* (ISM) [9], which constituted the basis for several extensions in the following years [10]. The ISM represents objects as a collection of a potentially large number of prototype patches, that in general exhibit a much denser coverage of the object area.

The ISM requires bounding box annotated training data to learn the model. The first step is to build a visual codebook, representing prototypical patch appearances. For each visual word the likelihood of having an object of the target class at the corresponding location is estimated by counting how often it is found on the object versus in the background. Additionally, for each occurrence on a positive training sample, the relative location of the object's centroid is stored. This information defines the spatial and the appearance model and is used to localize instances in previously unseen test images. During testing all local features are assigned to the most similar visual word. All object features vote for the object centroid locations and these votes, weighted by their foreground probability, are finally analyzed for providing detection hypotheses.

Despite the simplicity of this approach, the ISM has become one of the most popular object detection approaches due to some important properties. First, it has shown to yield excellent performance, mainly explained by the fact that a large number of local features contribute to each hypothesis. Second, it implicitly handles occlusions using a highly local and part based approach. Third, it has the possibility to combine parts from the whole range of positive training images and thus enables detection of object instance configurations never seen during training. Finally, inference is quite efficient since all features independently vote for the centroid. Variants of the ISM mainly differ in the way the visual vocabulary is built and how a local feature in an image is assigned to a visual word.

The final step of an ISM is to infer object location hypotheses from the provided centroid voting information. In this field mainly two dominant approaches emerged. The first approach, as proposed in [9], is based on applying mean shift over the voting space to estimate the probability density for the correct object lo-

cation. Afterwards, a Minimum Description Length (MDL) criterion is analyzed aiming at resolving ambiguities between neighboring hypotheses. The second approach is to accumulate the weighted centroid votes in a Hough voting manner (Generalized Hough Voting), where afterwards local maxima are identified by some type of non-maximum suppression, that discards the less confident of every sufficiently overlapping detection pair [5]. Although this summing of probabilities does not have a sound probabilistic interpretation, this simple accumulation process works quite well in practice.

All these methods have in common that they iteratively detect instances and they cannot enforce that patches are only assigned to a single object. Thus they all show quite limited performance if objects-to-be-detected are significantly overlapping. This was also pointed out in [11], where a probabilistic formulation of the object detection task based on the generalized Hough transformation was presented. This approach can be seen as a principled non-maximum suppression procedure, where the theoretical foundations for an improved analysis of the generalized Hough space were introduced. The algorithm has shown to especially improve results if detections overlap, as it frequently happens, *e.g.*, in human detection tasks.

In this work, we propose a novel random field based probabilistic formulation of the object detection task, based on the aforementioned ISM concept. Our method is most closely related to the work presented in [11], but, in contrast, we introduce a novel graph structure and adapt the semantics of the considered labels. This allows us to formulate our problem as a Markovian random field (MRF), which is one of the most popular models for structured inference. Considering object detection as application, we are able to define a novel semantic label space and a quite sparse graph structure. This structure enables a novel, efficient inference algorithm to search for a strong, local minimum of the random field energy function. In such a way, we find a common, global solution, where patches can be reassigned during the inference process. The final result of our method is a set of detection hypotheses, and for each detection the corresponding local patches that have voted for it. The proposed approach has several important properties: (a) we provide a joint, global solution for all object locations and individual pixel assignments, (b) we do not have to fix a range for local neighborhood suppression, (c) we maintain the results of standard NMS approaches in non-overlapping cases, while (d) implicitly separating even strongly overlapping object instances yielding (e) significantly improved detection scores.

2 A Random Field for object detection

The goal of this work is to provide a framework for object detection based on an Implicit Shape Model (ISM). Thus, as starting point, we assume that we are given a codebook consisting of several visual words, and that we can assign local features of a test image (*e.g.*, a dense set of patches) to the individual visual words. Additionally, each visual word stores a set of training samples (patches) that were assigned to it. These patches carry a label indicating if they

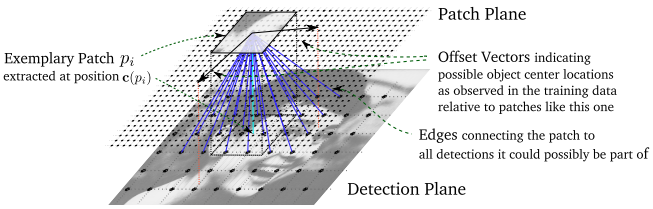

Fig. 1. Constructing a random field for object detection: The graph consists of a set of nodes positioned at patches extracted at each pixel in the image (the patch plane) and a coarser grid of nodes defining possible locations of detection centroids (detection plane). Each patch node is connected to the detection nodes it could be part of, where offset vectors stored in the implicit shape model define the connections' likelihood. Our novel inference process jointly solves the problems of detecting all objects and uniquely assigning contributing patch nodes to them.

appeared somewhere in the background (negative training set) or somewhere on a positive training sample. Those from positive training samples additionally store a relative offset vector to the corresponding object centroid. Given this information, our Implicit Shape Model is able to provide pixel-wise probabilities $p(y|x_i)$ for having a part of an object of category y at location i and a list of relative offset vectors to the object centroid.

The overall goal of our method is to fuse the provided information of the ISM in a probabilistically meaningful way, which jointly decides where in a test image instances of the learned category are depicted and which local features are part of the individual detections. Our method is based on a random field formulation. We first introduce the underlying graph structure in Sec. 2.1. An important part is our novel definition of a semantic label space tuned to the specific task of localizing objects based on an ISM, which is described in Sec. 2.2. Finally, we define our random field energy minimization problem in Sec. 2.3.

2.1 Two-layer graph structure

The core idea of this work is to take the probabilistic formulation of [11], and reformulate it to better fit the special case of object detection with Implicit Shape Models. One of the key insights of [11] was the following: an element in the ISM can vote for multiple objects at different positions in the image, because it was seen in training images on different locations relative to the object centroid. In one particular input image, however, each of the pixels is only part of exactly one object. Thus, when solving the detection task we ultimately have to decide for each patch to which detection it belongs (or implicitly do so). Contrary to the generic formulation in [11], we make use of the fact that in an ISM an element cannot vote for every detection hypothesis, but only for those that are reachable with an offset vector. The offset vectors define a fixed set of detection nodes a patch can interact with, relative to its position.

We thus define a two-layer graph structure, as illustrated in Fig. 1. The Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set of nodes $\mathcal{V}$ and edges $\mathcal{E}$ connecting the nodes. The

set of nodes $\mathcal{V}$ consists of patch nodes $\mathcal{P}$ at an image layer and detection nodes $\mathcal{D}$ at a detection layer (*i.e.*, $\mathcal{V} = \mathcal{P} \cup \mathcal{D}$). The patch nodes $\mathcal{P} = \{p_0, \ldots, p_{wh}\}$ form a grid spanning the whole input image, with one node per extracted, local feature for the ISM. In our case, this is the dense grid of pixels of the input image, *i.e.*, w, h are width and height of the image. The set of detection nodes $\mathcal{D} = \{d_0, \ldots, d_{uv}\}$ defines a coarser grid of size $u \times v$, where each d_j specifies the center of a potential object detection.

Each patch node p_i is connected by an edge $e_{p_i,d_j} \in \mathcal{E}$ to every node d_j that defines a detection that p_i could potentially be part of. This means, if the pixel coordinates of patch p_i, which we will denote as $\mathbf{c}(p_i)$, lie within a hypothetical detection bounding box centered at $\mathbf{c}(d_j)$, then there is an edge e_{p_i,d_j} connecting them. This is illustrated in Fig. 1 for one exemplary patch node.

Note that in this graph there are no connections between detection nodes. We initially intended to add such relations to implement local neighborhood suppression, but found that this is unnecessary in our framework. Since patch nodes are not allowed to contribute to more than one detection in the inference process, stronger detections pull away evidence from nearby detections automatically. Thus, our method does not require to fix a range for local neighborhood suppression as necessary in non maximum suppression methods, but implicitly is capable to separate even strongly overlapping object instances.

Using the graph $\mathcal{G}$, we define the random field, by associating a random variable with each node (which we also denote as p_i and d_j for simpler notation). Each random variable can be assigned one of the labels of the label set $\mathcal{L} = \{l_{\text{bg}}, l_{\text{fg}}, l_0, \ldots, l_n\}$. We will denote the label currently assigned to node v as l_v and the set of assignments to all patch and detection nodes as $l_{\mathbf{p}}$ and $l_{\mathbf{d}}$ respectively.

2.2 Defining the label set

The semantics of assigning one of the labels to a node, which is the essential characteristic of our formulation, is defined in the following way. Assigning the background label l_{bg} to a detection node (*i.e.*, $d_j = l_{\text{bg}}$) means that there is no detection at this position. Likewise, a configuration having $d_j = l_{\text{fg}}$ specifies that there is an object centered at $\mathbf{c}(d_j)$. For a patch node $p_i = l_{\text{bg}}$ signifies that at the center of the patch $\mathbf{c}(p_i)$ there is no object, but background. This does not imply that none of the detection nodes connected to p_i can be set to l_{fg}, since the bounding box of a detection might well contain some background pixels.

The crucial point of our framework is the meaning of the labels $l_0, \ldots, l_n$. Assigning one of these labels to a patch node indicates that this patch is part of a detection centered on a specific detection node, specified as follows. As shown in Fig. 2, the detection node with the closest pixel coordinates to the patch (printed in dark blue) defines the origin of a coordinate system of relative offsets in the detection grid. From the training data, we can determine the maximal range of the offset vectors, stored with the codebook entries. This range defines a fixed rectangular area of detection nodes that a patch could potentially vote for with its offset vectors. Within this area we reserve a separate label for each detection

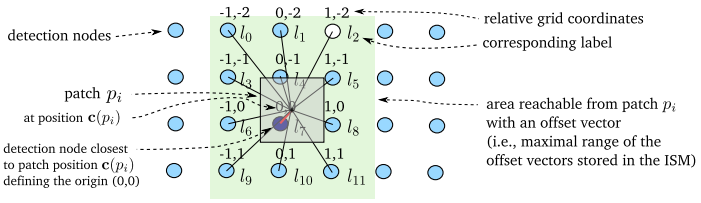

Fig. 2. A patch p_i centered at pixel $\mathbf{c}(p_i)$ is connected to all detections it could potentially be part of. The relative position on the grid defines the semantics of the labels for this patch. *E.g.*, assigning label l_2 to the patch would mean that it is part of (votes for) a detection centered at the detection node (white) at position (1,-2) relative to its closest detection node (dark blue).

node. Assigning this label to the patch means that it is part of the corresponding detection. For an example, see Fig. 2. Note that the set of labels is the same for all patch nodes. However, the semantic meaning of label assignments is spatially varying, since the label implicitly defines an assignment to different object hypotheses depending on the location of the patch. For notational convenience, we will denote the label that specifies that the patch at p_i is part of the object centered on detection node d_j as $\hat{l}_{i,j}$.

2.3 Energy function

Given an input image I and the graph structure as defined in Section 2.1, the probability of an assignment of labels to all nodes, *i.e.*, a total configuration of the random field, can be written as

$$p(l_{\mathbf{p}}, l_{\mathbf{d}}|I) = \prod_{p_i \in \mathcal{P}} p(l_{p_i}|I) \prod_{d_j \in \mathcal{D}} p(l_{d_j}|I) \prod_{e_{p_i,d_j} \in \mathcal{E}} p(l_{p_i}, l_{d_j}|I)\,. \tag{1}$$

Taking the log of Eq. (1) leads to the formulation of the energy function to be minimized:

$$E(l_{\mathbf{p}}, l_{\mathbf{d}}) = \sum_{p_i \in \mathcal{P}} \psi_{p_i}(l_{p_i}) + \sum_{d_j \in \mathcal{D}} \psi_{d_j}(l_{d_j}) + \sum_{e_{p_i,d_j}} \psi_{i,j}(l_{p_i}, l_{d_j})\,, \tag{2}$$

where $\psi_{p_i}(l_{p_i}) = -\log(p(l_{p_i}|I))$ is the unary cost of assigning the label l_{p_i} to node p_i, $\psi_{d_j}(l_{d_j})$ is the equivalent for detection node d_j and $\psi_{i,j}(l_{p_i}, l_{d_j})$ is the resulting pairwise cost. With these definitions, finding the objects in the image amounts to finding the assignment of labels to all nodes that minimizes Eq. (2).

Definition of Unary Potentials. Starting with the first term in Eq. (1), $p(l_{p_i}|I)$ represents the probability of assigning the label l_{p_i} to node p_i, given the image data. Let x_i be the appearance of the local feature extracted around $\mathbf{c}(p_i)$. By making the same independence assumption as in [11], namely that the probability of a label on a patch only depends on its appearance x_i, we can define the

posterior probability of the labeling of a patch node by $p(l_{p_i}|I) = p(l_{p_i}|x_i)$. This probability can be derived from the statistics collected in the ISM as follows.

In order to get an estimate of how likely a detection at a certain position is, given one patch, we have to sum up the offset vectors that point from the patch to that detection. Since, as in [5], we want to allow the patch to move slightly around its original offset position, all voting vectors are aggregated by a Gaussian centered at the detection.

This summing up of evidence for an object center around a detection node also has a different interpretation. In order to achieve the tolerance for small shifts of the patch, we could also resample the training set and insert additional offset vectors pointing to positions around the original centroid location, giving the same effect as the smoothing with a Gaussian. Unfortunately, this smoothing or resampling introduces additional virtual samples that change the ratio of positive to negative samples in the ISM statistics. Thus, it is not possible to directly take the summed up voting weights at each detection node as probabilities for the labels. Correcting for this bias would be a tedious task since the amount of virtually introduced samples depends on the density of the detection grid and the distribution of offset vectors. Additionally, the statistics stored with the codebook are not completely reliable, as for instance an entry with no single negative training patch would indicate zero probability for a patch with this appearance to appear somewhere in the background. This almost certainly does not reflect truth but is an artifact of insufficient training data.

Barinova *et al.* [11] bypass these problems, by setting the probability for assigning background to a patch node to a constant chosen on a validation set. We take a different approach, trying to make more use of the inexact but nonetheless valuable information stored with the codebook entries. We take the probability of being foreground (pfg_{p_i}) estimated from the original ratio of training samples stored in the ISM and estimate $p(p_i = l_{\mathrm{bg}}|x_i)$ by taking it as input to the shifted sigmoidal function:

$$p(p_i = l_{\mathrm{bg}}|x_i) = 1 - \frac{pfg_{\max}}{1 + \exp(-\alpha(pfg_{p_i} - \beta))}\,. \tag{3}$$

All parameters of this function can be estimated once on a validation set and are kept fixed at $pfg_{\max} = 0.95, \alpha = 10, \beta = 0.4$. This procedure of limiting the foreground probability to a maximum value of $pfg_{\max}$ can also be seen as combining the estimated distribution with a uniform Dirichlet prior. The probabilities for the labels $l_0, \ldots, l_n$ are then defined by taking the evidence gathered above for each detection node and scaling it such that the maximum reaches $1 - p(p_i = l_{\mathrm{bg}}|x_i)$.

The second term of Eq. (1), $p(l_{d_j}|I)$, encodes the probability for a label on a detection node. This can be used to express a prior probability for a detection. However, in practice we do not make assumptions about the distribution or frequency of detections and thus set $p(d_j = l_{\mathrm{bg}}) = p(d_j = l_{\mathrm{fg}}) = 0.5$. Detection nodes can thus be seen as auxiliary variables, collecting the information of its connected patch nodes via the pairwise relations. All other labels $l_0, \ldots, l_n$ are invalid for detection nodes, so their probability is set to 0.

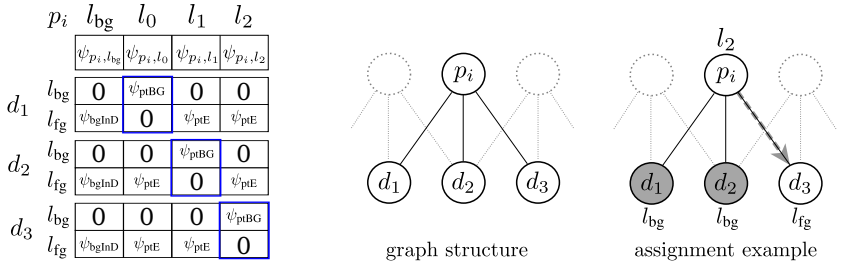

Fig. 3. A simple 2D example: each patch node connects to three detection nodes. The table on the left shows all unary (first row) and binary costs for all possible labeling combinations for one patch node p_i and its associated detection nodes. Example shown on the right: Let $p_i = l_2$ (*i.e.*, patch p_i votes for detection d_3) and the detection nodes are set to $d_1 = l_{\text{bg}}, d_2 = l_{\text{bg}}, d_3 = l_{\text{fg}}$; the total cost of the configuration is $\psi_{p_i}(l_2) + \psi_{i,1}(l_2, l_{\text{bg}}) + \psi_{i,2}(l_2, l_{\text{bg}}) + \psi_{i,3}(l_2, l_{\text{fg}}) = \psi_{p_i,l_2} + 0 + 0 + 0$.

Definition of Pairwise Potentials. The pairwise costs $\psi_{i,j}(l_{p_i}, l_{d_j})$ reflect the semantics of the labels for the relationship between patch nodes and detection nodes. Fig. 3 shows a simple example with one exemplary patch node connected to three detection nodes. The tables on the left list the costs for all different kinds of label configurations for one patch and its neighboring detection nodes. The first row contains the unary cost for assigning each label to the patch node $\psi_{p_i}(l_{p_i})$, as defined above. The three separate tables below show the pairwise costs for combinations of label assignments to the patch and each detection node. Each has one row for the costs of assigning l_{bg} and l_{fg} to the detection node respectively. The blue frames mark the column with the costs for assigning label $\hat{l}_{i,j}$ to the patch, which means that the patch p_i is part of the corresponding detection d_j.

If detection and patch are both assigned to background, this is a valid combination and the cost is 0. The same is true if a detection $d_j = l_{\text{bg}}$ and the patch is set to anything else but $\hat{l}_{i,j}$, or $d_j = l_{\text{fg}}$ and $p_i = \hat{l}_{i,j}$ (*i.e.*, the detection is switched on and the patch is part of it). A patch being part of a detection at an inactive detection node (*i.e.*, $p_i = \hat{l}_{i,j} \wedge d_j = l_{\text{bg}}$) is an invalid configuration resulting in a cost of ψ_{ptBG}, which we can set to ∞ (or in practical implementations to a very high cost). Conversely, a patch assigned to background $p_i = l_{\text{bg}}$, in the range of an active detection $d_j = l_{\text{fg}}$, adds a fixed cost ψ_{bgInDet}, derived from the probability that a pixel inside a detection rectangle might be background, which can be estimated from the training data. This expresses the fact that objects in the training and test data do not completely fill the bounding box they are annotated with. This parameter also controls how much of an object must be visible (not occluded) for a valid detection. Finally, from the point of view of the detection, there is no difference if the patch is assigned to background or to any other detection close by, so $\psi_{\text{ptE}} = \psi_{\text{bgInDet}}$.

3 Inference

Since our defined pairwise costs fulfill the conditions of *regularity* [12], we could, *e.g.*, apply standard graphcut-based inference methods such as alpha expansion or alpha/beta swap [13] to solve the labeling problem. However, generic solving algorithms fail for our particular graph structure and definition of potentials. The main problem is that trying to change a single node, or even all nodes, to exactly one new label can almost never result in a lower energy.

For example, as can be seen from Fig. 3, if all nodes are assigned the background label l_{bg}, switching a patch node to any different label will result in adding the very high cost ψ_{ptBG} at one binary relation. Switching a single detection node to l_{fg} will not change the unary cost for this node but increase the total energy of each pairwise edge that connects this detection node to any patch node by the cost ψ_{bgInDet}. Thus, setting every node to l_{bg} results in a strong local minimum of the energy and thus, inference approaches like alpha-expansion that only consider changing nodes to a single, new label per iteration, immediately fail.

For this reason, we propose an inference approach tuned to our specific graph structure and label semantics. The core idea is a novel move making strategy, which is described in detail in Section 3.1. The corresponding inference process is outlined in Section 3.2, while Section 3.3 discusses the overall characteristics of our inference approach.

3.1 Moves

We propose to use a different kind of move, specialized for our problem setup, that changes the labels of several nodes simultaneously. The central observation, that was also already pointed out in [11], is that given a labeling of the detection nodes, the optimal label for each patch can be determined independently, since the graph is bipartite. Careful inspection of our setup reveals that we can efficiently compute the new optimal assignment for each patch node when a single detection node changes its label in $O(1)$, if we know the previously optimal assignment and cost. This allows us to construct an efficient inference algorithm that is described in the following paragraphs.

The prerequisite of a starting point with known optimal assignments of the patch nodes and total costs is easily fulfilled by setting all detection nodes to l_{bg}. The optimal label for each patch is then also l_{bg}, because any other label would add ψ_{ptBG} to the total cost. The total energy of this configuration amounts to the sum of unary costs for l_{bg} of all detection and patch nodes (all pairwise costs are 0). The sum of costs for each label at each patch node, that we will need later in the process, is its unary cost plus, for each label other than l_{bg}, one binary cost of ψ_{ptBG}.

Then, we consecutively turn on one detection d_j after the other, and find the optimal configuration of patch node labels for the new situation, to discover which one lowers the total energy most. To compute the total energy of a new configuration we need to keep track of the change of energy ΔE for each node

that changes its label during the process, and affected edges. The change in unary cost for the detection node is $-\psi_{d_j}(l_{\text{bg}}) + \psi_{d_j}(l_{\text{fg}})$. Since none of the connected patch nodes can have pointed to d_j before (since it was background), all pairwise relations switch from 0 cost to ψ_{bgInDet} or ψ_{ptE} (which are equal).

Now we have to check for each patch node p_i connected to the currently tested detection node d_j, for the new best label. The optimal label for a patch node depends on the patch's unary cost for the label, plus the pairwise to all detection nodes it is connected to. To find the label with lowest energy in a brute force manner, we would have to go over all labels and for each of them sum up the binary costs of all the edges of the patch. Despite our sparse graph structure in which a patch is not connected to all detection nodes, this would require $O(|L|^2)$. But, in fact, for every label the only change in energy is in the pairwise connection between the patch and the changed detection d_j, so we can update them incrementally. Additionally, we do not even have to go over all labels to find the new best, since we know that the label currently assigned to p_i was the best one before the current move and looking at Fig. 3 we see that only switching to $\hat{l}_{i,j}$ can possibly result in a decrease of the energy. So the only possibility we have to check is, if the total cost of the patch's old label is now bigger than the cost for $\hat{l}_{i,j}$. We keep track of the total change of costs for the better of those two possibilities. This is an $O(1)$ operation for each patch.

3.2 Overall inference process

Thus, in total, calculating the change in energy for switching on a single detection hypothesis and finding the optimal configuration of patch labels is a fast operation. Therefore, we can afford to test every single detection hypothesis and take the best one, without having to rely on a heuristic to propose potentially good hypotheses. After the new best detection hypothesis is found, we switch the corresponding detection node d_j to l_{fg} and each patch node, for which this results in a better energy, to $\hat{l}_{i,j}$, to associate it with the new detection, and update the costs for each label. This is only done once per newly found detection and only for the patches connected to the new detection. The whole process is repeated until no move lowers the total energy.

3.3 Discussion

Note that the decision of finally taking the most probable detection in each iteration is greedy. However, the greedy decision is based on the evaluation of every single possible move that switches on one detection node and finds the new optimal configuration of all patch nodes. Even after the move is taken, the patch nodes that were switched to the new detection in this iteration, are not fixed to this decision, but can switch to a different detection found later, if this again decreases the total energy.

Additionally, the order of configurations checked by the algorithm assures fast convergence to a good minimum of the energy by making use of domain

knowledge. For instance, in a generic solver it would be hard to exploit the fact that switching on lots of detection nodes at once is very unlikely to give a low energy, or encode the knowledge which set of labels to apply to the corresponding patch nodes.

As a final remark we would like to point out that after the first iteration not all possible remaining hypotheses have to be checked again to find the next best detection. The total benefit (reduction of cost) for each hypothesis can only become smaller with the new detection from the last iteration now switched on and adding pairwise costs to every patch node not pointing towards it. Thus, we do not have to check those hypotheses that already did not have a negative ΔE in the last run. Since even for a crowded scene the number of objects is way lower than the number of detection nodes, this again dramatically reduces the search space.

4 Experiments

To create the codebook for the ISM, we use Random Forests, trained as proposed in [5]. The smoothing kernel's sigma is set to $\sigma = 3.0$ to allow for small shifts of the patches, with respect to the object center. Derived from this, we set the resolution of the detection grid to 8×8. A coarser grid would miss detections, because the patches can only vote for detection sites within the range of the Gaussian. A denser grid would linearly increase computation time with the number of detection nodes. The only parameter left to set is ψ_{bgInDet}. Basically, it defines how much of an object must be visible in order to create a positive detection. Since we want to detect highly overlapping instances, we set it quite low, to a value of 0.4. Conversely this implies a high probability of about $e^{-0.4} \approx 67\%$ of a patch to be background within a valid detection.

To get multi-scale detection results, we first process each scale individually, but afterwards, according to our localization principle, ensure that also over scales each patch only votes for a single detection. From the final configurations of the random fields per scale, we obtain all detections and the corresponding set of patches assigned to them, which defines a pixel-wise voting mask per detection (see Fig. 5). We collect these masks over all scales and resize them to a reference frame. Then we sort all detections by their confidences and, starting with the most confident, accept only detections that do not overlap (considering the voting masks) with those already taken. Thus, we again ensure that also over scales each patch is only assigned to a single detection. Thereby, we effectively suppress lower scoring redetections in nearby scales and obtain a unified solution for multi-scale analysis.

Similar to [5], we report rectangles of mean aspect ratio (estimated from the training data) centered at each active detection node. The confidence of each reported detection is set to the absolute value of the decrease in energy that was recorded during testing the corresponding detection node.

4.1 Datasets

The choice of evaluation datasets is motivated by several factors. First we want to have a direct comparison to the most closely related approach [11]. The publicly available implementation comes with its own set of random forests, trained for detection of side views of pedestrians. Thus, we also focus on this task, although our method is not specifically tailored towards it and potentially handles arbitrary object categories.

Another aspect is the resolution of the objects in the images. Part based approaches, like ISMs, can only capitalize on their strengths if the objects are depicted at a resolution where the parts are distinguishable. Thus we require the smallest category instances to have at least about 100 pixels in height.

Since for non overlapping object instances our proposed method reaches the same decisions as standard NMS (as was tested and assured in evaluations on single scale datasets like UIUC cars), we are especially interested in testing the capability of our algorithm to resolve detections of strongly overlapping objects. Thus, we evaluate it on the *TUD crossing* and *TUD campus* sequences [14], also used in [11], where both datasets require the ability to locally decide for each patch to which detection it belongs in a reasonable manner. Additionally, we evaluate all approaches on the PETS 2009 dataset, also featuring close to side views of a large number of pedestrians with heavy overlaps.

The *TUD campus* and *TUD crossing* datasets contain two sequences of images showing pedestrians in street side scenarios. In *TUD campus* there are 71 images of highly overlapping persons walking along a side walk. This sequence especially features large changes in scale, as well as strong occlusions. The *TUD crossing* sequence contains 201 images of a relatively crowded scene, showing profile views of pedestrians crossing a street. We use the extended ground truth from [15].

The PETS 2009 Benchmark Data [16] includes several datasets of which we take View001 of sequence S1.L1 to further evaluate the performance of our method. This sequence shows several groups of people walking closely and thus heavily occluding each other. In total the groundtruth annotation contains 4348 persons.

As training data for all experiments we use the training set of the *TUD-pedestrian* dataset, consisting of 400 images of mostly side views of pedestrians.

4.2 Results

We directly compare our results to the two most related approaches: the Hough Forests using standard non maximum suppression [5] and the probabilistic framework of Barinova *et al.* [11]. Detections are considered as valid analyzing the standard PASCAL-VOC overlap criterion, with the threshold set to 50%. For both methods compared, we used the publicly available source codes and associated configuration files as published by the respective authors. For training the Hough Forests we used exactly the same data as for our method.

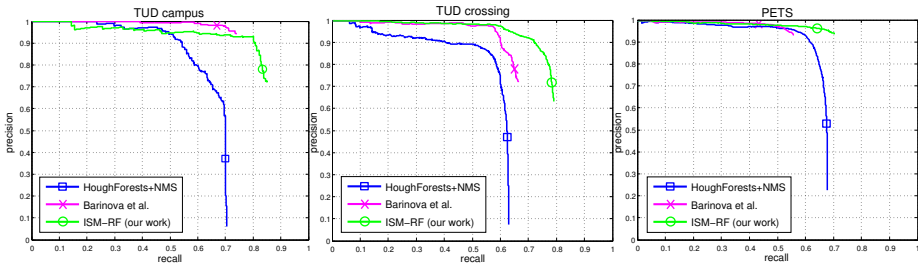

Fig. 4. Precision/Recall curves on (a) *TUD campus*, (b) *TUD crossing* and (c) PETS 2009 S1.L1 sequence for all three methods.

Fig. 5. Sample detections on the TUD crossing sequence (top row) and PETS 2009 (bottom row). For each detection the uniquely assigned patches are plotted in a different color. Note how closely walking pedestrians, overlapping each other, are correctly separated.

Fig. 4 shows precision-recall curves for all three methods on all three databases. As can be seen, our method significantly improves over [5] and also outperforms [11] on all three datasets, gaining about 10% recall, at precision levels above 90%. Fig. 5 additionally visualizes detection results and all local patches that were assigned to each detected instance by our inference process in different colors. Note how even strongly overlapping persons are correctly separated from each other.

5 Conclusion

In this work, we have proposed a new formulation for the task of detecting objects based on the information gathered in an Implicit Shape Model. We formulated the dual problem of detecting a set of object hypotheses and assigning

local patches to the detections in a random field manner using a significantly sparser graph structure than in related approaches. Furthermore, the specific graph structure allowed to define a novel, fast inference algorithm to solve our defined energy minimization problem. Our method does not require to fix a range for local neighborhood suppression as it is necessary in related methods, but implicitly is capable to separate even strongly overlapping object instances. Experiments demonstrated that we are able to accurately detect object hypotheses and their local support patches on challenging data sets achieving competitive or even improved results in comparison to state-of-the-art in this field.

Acknowledgement. This work was supported by the Austrian Science Foundation (FWF) project Advanced Learning for Tracking and Detection in Medical Workflow Analysis (I535-N23) and by the Austrian Research Promotion Agency (FFG) project SHARE (831717) in the IV2Splus program.

References

1. Viola, P., Jones, M.: Robust real-time face detection. IJCV **57** (2004) 137–154
2. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE Trans. on PAMI **31** (2009) 2129–2142
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR. (2005)
4. Fischler, M., Elschlager, R.: The representation and matching of pictorial structures. IEEE Trans. on Computers **C-22** (1973) 67 – 92
5. Gall, J., Lempitsky, V.: Class-specific Hough forests for object detection. In: Proc. CVPR. (2009)
6. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. on PAMI **32** (2010) 1627–1645
7. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scale-invariant learning of models for visual recognition. IJCV **71** (2007) 273–303
8. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. ICCV. (2003)
9. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Proc. ECCV. (2004)
10. Lehmann, A., Leibe, B., Gool, L.V.: PRISM: PRincipled Implicit Shape Model. In: Proc. BMVC. (2009)
11. Barinova, O., Lempitsky, V., Kohli, P.: On the detection of multiple object instances using Hough transforms. In: Proc. CVPR. (2010)
12. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? IEEE Trans. on PAMI **26** (2004) 147–159
13. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on PAMI **23** (2001) 1222–1239
14. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: Proc. CVPR. (2008)
15. Riemenschneider, H., Sternig, S., Donoser, M., Roth, P.M., Bischof, H.: Hough regions for joining instance localization and segmentation. In: Proc. ECCV. (2012)
16. Ferryman, J., Shahrokni, A.: PETS2009: Dataset and challenge. In: PETS. (2009)