Structured Labels in Random Forests for Semantic Labelling and Object Detection

Peter Kontschieder, Samuel Rota Bulò, Marcello Pelillo, *Fellow, IEEE*, and Horst Bischof

Abstract—Ensembles of randomized decision trees, known as Random Forests, have become a valuable machine learning tool for addressing many computer vision problems. Despite their popularity, few works have tried to exploit contextual and structural information in random forests in order to improve their performance. In this paper, we propose a simple and effective way to integrate contextual information in random forests, which is typically reflected in the structured output space of complex problems like semantic image labelling. Our paper has several contributions: We show how random forests can be augmented with structured label information and be used to deliver structured low-level predictions. The learning task is carried out by employing a novel split function evaluation criterion that exploits the joint distribution observed in the structured label space. This allows the forest to learn typical label transitions between object classes and avoid locally implausible label configurations. We provide two approaches for integrating the structured output predictions obtained at a local level from the forest into a concise, global, semantic labelling. We integrate our new ideas also in the Hough-forest framework with the view of exploiting contextual information at the classification level to improve the performance on the task of object detection. Finally, we provide experimental evidence for the effectiveness of our approach on different tasks: Semantic image labelling on the challenging MSRCv2 and CamVid databases, reconstruction of occluded handwritten Chinese characters on the Kaist database and pedestrian detection on the TU Darmstadt databases.

Index Terms—Random forests, structured prediction, semantic image labelling, object detection

1 Introduction

The importance of context in human visual processes and in our everyday judgements can hardly be exaggerated. It is a common-sense observation that objects in the real world do not live in a vacuum, and researchers have gone so far as to maintain that all attributions of knowledge are context-sensitive, a view commonly known as contextualism [1], [2]. The use of contextual constraints in pattern recognition dates back to the early days of the field, especially in connection to optical character recognition problems [3], [4], and reached its climax within the computer vision community in the 1980s with the development of relaxation labelling processes [5], [6] and Markov Random Fields (MRF) [7]. Recently, the computer vision community is again paying increasing attention to the role played by contextual information in visual perception, especially in high-level problems like object recognition or semantic scene segmentation.

The goal of our work is to exploit structured, contextual information from the labelling output space within the random forest framework [8], [9], [10] in order to improve on semantic image labelling and object detection tasks. Random forests are ensembles of randomized decision trees (DTs) which are considered to be competitive to other state-of-the-art learning techniques such as boosting or support vector machines (SVMs) and have successfully enabled many applications [11]. They exhibit several appealing properties: They are extremely fast for training and classification, can be easily parallelised [12], are inherently multi-class capable, tend not to overfit and are robust to label noise [10]. In addition, they were shown to outperform other learning techniques on high-dimensional data problems and are close to an ideal learner [13]. We introduce them in general terms in Section 2 and in the context of computer vision applications in Section 2.1. Our main motivation is that random forests have become an effective tool for classification and regression in computer vision applications, but little has been done in the direction of learning contextual dependencies within this framework. Our work provides a novel solution in this respect, which is supported by an extensive experimental evaluation. We start with providing more details about our contributions in this work before we give a coarse overview on the use of context in computer vision in Section 1.2.

1.1 Contributions

In this work, we provide a novel approach for the exploitation of contextual information in random forests. We depart from the standard classification setting where a single (atomic) label is associated to each training sample, and instead take structured label information from each pixel's neighbourhood into account. As an example, consider the problem of classifying pixels into semantic categories as shown in Fig. 1. On the bottom left, we show some training samples that can be used to train a standard random forest.

- *P. Kontschieder is with the Machine Learning and Perception group at Microsoft Research, Cambridge, UK. E-mail: pekontsc@microsoft.com.*
- *S. Rota Bulò is with ICT-TeV, Fondazione Bruno Kessler, Trento, Italy. E-mail: rotabulo@fbk.eu.*
- *M. Pelillo is with DAIS, Università Ca' Foscari Venezia, via Torino 155, Venezia Mestre 30172, Italy. E-mail: pelillo@dsi.unive.it.*
- *H. Bischof is with the Institute for Computer Graphics and Vision, Graz University of Technology, Graz A-8010, Styria, Austria. E-mail: bischof@icg.tugraz.at.*

Manuscript received 14 Apr. 2013; revised 17 Jan. 2014; accepted 16 Mar. 2014. Date of publication 6 Apr. 2014; date of current version 10 Sept. 2014. Recommended for acceptance by A. Torralba.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2014.2315814

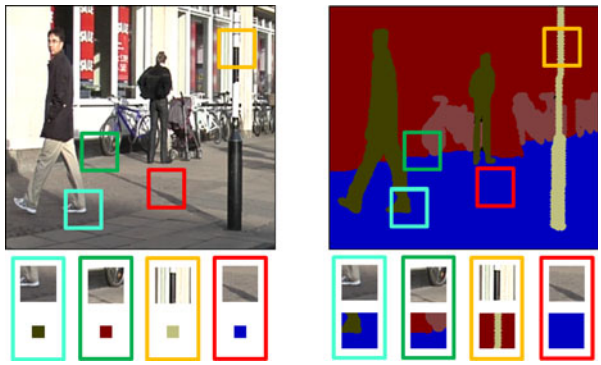

Fig. 1. Comparison of atomic labels and structured labels. Top: Training image and related label image with colour-coded patch samples. Bottom: Associated, atomic labels and structured labels.

Each sample consists of a patch of the image centred on a pixel and the associated single, atomic class-label. On the bottom right, we can see instead our idea of incorporating structured information by retaining a whole patch of labels centred on each sample. In such a way, we are moving from an unstructured to a *structured* output space—A transition we describe in detail in Section 3.1.

From a methodological point of view, we show how random forests can be adapted in order to take advantage of this additional information. In Section 3.2, we provide a new test function selection criterion which allows to exploit the information of structured label patches in the tree growing procedure. In Section 3.3 we show how our forests provide structured predictions for each pixel of a test image and propose different mechanisms to translate them into a coherent, final labelling result. As additional contribution, we extent our idea to also account for object detection tasks by integrating a generalized Hough transform into our forest model. The approach follows [14] with the important difference that contextual information about the labelling is exploited during training. Details can be found in Section 4.

From the experimental perspective, we show that by including contextual information at the classification level within the random forest, the results drastically improve for applications like semantic image labelling and object detection. Indeed, the structured output space allows to counteract the assignment of meaningless label configurations, as experienced when using standard classification forests. This is due to the fact that our forest can learn the label topology characterizing the ground-truth labellings and avoid implausible label transitions during the inference stage. Section 5 is devoted to the experimental evaluation of the proposed approach. In Sections 5.1 and 5.2 we perform experiments on several semantic image labelling databases like Cambridge-driving Labelled Video Database (CamVid) and MSRCv2, where the goal is to assign a semantic category to each pixel of an image. In Section 5.3 we also address the problem of inpainting by reconstructing occluded handwritten Chinese characters. Sections 5.4, 5.5, and 5.6 are devoted to evaluating the performance of our forest on the task of pedestrian detection on different databases. Finally, we want to mention that preliminary versions of this work appeared in [15] and [16].

1.2 Literature Review

Early efforts to exploit context in computer vision considered contextual dependencies as expert knowledge from a priori information [17], [18], [19]. With the introduction of *relaxation labelling processes* [5], contextual information was included to provide more consistent solutions in classification problems where noise and uncertainty typically affected the accuracy of classical, non-contextual pattern recognition algorithms [6], [20], [21]. Later, *random field* [7], [22] based approaches appeared in the computer vision field and led the way to the use of *graphical models* for modelling contextual relations.

A few works are based on *directed graphical models* [23], [24], [25], while the majority exploits *undirected graphical models*, such as Markov Random Fields [26] or Conditional Random Fields (CRF) [27], [28], [29], [30]. The former models assume the existence of a latent causal process that produced the observed image, while the random field models are better suited to handle soft constraints between image components with no natural causal relationship among them. In the random field models, context dependencies are mostly provided in terms of higher-order parametric energies, which allow to specify spatial and semantic constraints at the pixel-, object- and scene level.

Recent state-of-the-art approaches [31], [32], [33] typically incorporate complementary features at different levels. Low-level features are mostly calculated on a per-pixel basis and incorporate local colour or texture statistics or outputs of weak classifiers, while mid-level features operate on regions or superpixels to provide shape, continuity or symmetry information. Motivated by perceptual psychology [34], high-level features introduce global image statistics and information about inter-object or contextual relations, seeking for proper scene configurations on the image level. Interesting developments of the random field model are the *Decision Tree Fields* (DTF) [35], where a conditional random field instance is generated for each test image under the guidance of trained decision trees, holding the CRF parameters. Further improvements of this work can be found in [36].

Probabilistic graphical models, and in particular random fields, fall within the broader class of structured prediction models. Embedded in the context of structured learning theory, we have recently introduced a different model in [37]. There, we can learn contextual dependencies using so-called *Structured Local Predictors*, i.e., parametrized functions that determine the class prediction of each pixel as a function of relative position, appearance and class of neighbouring pixels. In [38] an approach to semantic segmentation is proposed that tries to fit templates of label configurations at different levels of details in order to capture long-range dependencies and exploit different levels of contextual information. Moreover, the work in [39] introduces the concept of *Local Label Descriptors*, guiding the alignment process of label patches, which in turn are forced to mutually agree and correlate with the local feature descriptors within an image. For more information on structured learning and prediction in computer vision we refer to the comprehensive tutorial of [40].

A different way to learn a contextual model was presented in [41] with an approach named *Auto-context*. There,

a sequence of classifiers were trained in a boosting-fashion using both appearance-based features and contextual information obtained by the classifiers from previous training iterations. However, the reported learning phase is computationally very demanding. Other approaches like boosted random fields [42] or SpatialBoost [43] share both the disadvantage of significant computational complexity when considering contextual beliefs as weak learners. Similar to [41], the *Texton Forest* of [44] introduced a model using context information, but for the first time in the random forest framework. In [45], the authors propose *Entangled Decision Forests* for semantic image labelling, which integrate the auto-context model in random forests by exploiting the decision tree hierarchy. Recently, we have introduced a related approach in [46], where we jointly perform regression and classification for the task of object detection. In this line, [47] has shown how to learn contextual information by exploiting geodesic connectivity features over the space of semantic class predictions together with introducing a random-field like training objective for the internal tree split nodes. The introduction of a preliminary version of this work in [16] has triggered more research into this direction. For instance, [48] has formulated the task of predicting local edge masks as structured output prediction problem in random forests, using an intermediate mapping of structured labels onto a discrete space where conventional information gain analysis can be undertaken. At this point we also refer to [49], where image categorization with random forests was performed as structured input space analysis.

2 Randomized Decision Forests

We start by briefly revisiting the randomized decision forest and introducing some notations used in the subsequent sections. A (binary) *decision tree* is a tree-structured classifier which makes a prediction by routing a feature sample $x \in \mathcal{X}$ through the tree to a leaf, where the classification takes place. Each leaf stores a class label $\pi \in \mathcal{Y}$ which is associated to any sample reaching it.[1] At each internal node, a sample $x \in \mathcal{X}$ is forwarded to the left or to the right based on the output of a node-specific split function $\psi : \mathcal{X} \rightarrow \{0, 1\}$.

A *decision forest* is an ensemble $\mathcal{F}$ of decision trees which makes a prediction about a sample by combining the single predictions collected from the trees in the ensemble. Formally, the prediction for sample x from a forest $\mathcal{F}$ is the class label $y^* \in \mathcal{Y}$ receiving the majority of the votes among the trees, i.e.,

$$y^* \in \arg\max_{y \in \mathcal{Y}} \sum_{t \in \mathcal{F}} \mathbb{1}[h(x \,|\, t) = y], \tag{1}$$

where $h(x \,|\, t) \in \mathcal{Y}$ denotes the prediction for sample x obtained from tree $t \in \mathcal{F}$, and $\mathbb{1}[\cdot]$ is the indicator function returning 1 or 0 according to whether the passed argument is true or false.

We train a binary decision forest similar to the randomized trees algorithm in [10]. Each tree in a forest is trained independently on a random subset of the training set $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ according to a recursive learning procedure. There are several criteria to terminate this recursion, i.e., a leaf node is grown when: $|\mathcal{D}|$ is smaller than a minimum size or if the entropy of its class distribution $\mathbb{H}(\mathcal{D})$ is below a given threshold. Also the maximum depth (i.e., the number of preceding non-leaf nodes) of the tree may be used as termination criterion. If one or several of the termination criteria hold, a leaf is grown whose class prediction π is set to the most represented class in the training data $\mathcal{D}$, i.e.,

$$\pi \in \arg\max_{z \in \mathcal{Y}} \sum_{(x,y) \in \mathcal{D}} \mathbb{1}[y = z]\,. \tag{2}$$

Otherwise an internal node is grown with an associated split function ψ selected from a randomly generated set Ψ in a way to maximize the expected information gain about the label distribution due to the split $\{\mathcal{D}_l^{\psi}, \mathcal{D}_r^{\psi}\}$ of the training data, induced by ψ. Formally, ψ is the solution of the following minimization:

$$\psi \in \arg\min_{\psi' \in \Psi} \Big\{ |\mathcal{D}_l^{\psi'}| \mathbb{H}(\mathcal{D}_l^{\psi'}) + |\mathcal{D}_r^{\psi'}| \mathbb{H}(\mathcal{D}_r^{\psi'}) \Big\}\,. \tag{3}$$

Finally, the node's left and right sub-trees are recursively grown with their respective training data $\mathcal{D}_l^{\psi}$ and $\mathcal{D}_r^{\psi}$.

In case of unbalanced training data among the different classes to be learned, the tree classifiers can be trained by weighting each label according to the inverse class frequencies observed in the training data. The weights are also considered in the computation of the expected (weighted) information gain, which determines the selection of the best test function during the training procedure. Consequently, the prediction error on the class average score is reduced as opposed to the global score in the non-rebalanced case.

2.1 Random Forests in Computer Vision

Recently, random forests were customized for a large variety of tasks in computer vision [14], [50], [51], [52], [53], [54]. For classification tasks in the image domain, the feature space is typically anchored to a pixel grid topology such that random forests can be trained on a specific feature space $\mathcal{X}$, which consists of a set patches extracted from a set of multi-channel images $\mathcal{I}$, where channels may include colour features such as gradients, filter banks, etc.

More formally, a training image with d channels is a function $I \in \mathcal{I}$ mapping pixels (i.e., elements of $\mathbb{Z}^2$) to d-dimensional feature vectors. The set of pixels composing image I is identified by the domain of I denoted as $\mathrm{dom}(I)$. The value of channel c at pixel $\mathbf{u} \in \mathbb{Z}^2$ in image I is written as $I(\mathbf{u})_c$. A patch is identified by a pair $(\mathbf{u}, I) \in \mathcal{X}$, representing the coordinates $\mathbf{u} \in \mathbb{Z}^2$ of the patch centre in image $I \in \mathcal{I}$. In this way, the patch shape is implicitly characterized by the type of test function. The label space $\mathcal{Y} = \{1, \ldots, k\}$ is given by the set of k object classes we are going to find in the images.

Different types of test functions for a patch $x = (\mathbf{u}, I) \in \mathcal{X}$ have been investigated for the classification task (see, e.g., [16], [44]). Test functions of random type and with randomly generated parameters are drawn during the training procedure to form the set Ψ of split functions in each node of the decision trees.

1. One could also store a probability distribution over the class-labels instead of a single class-label.

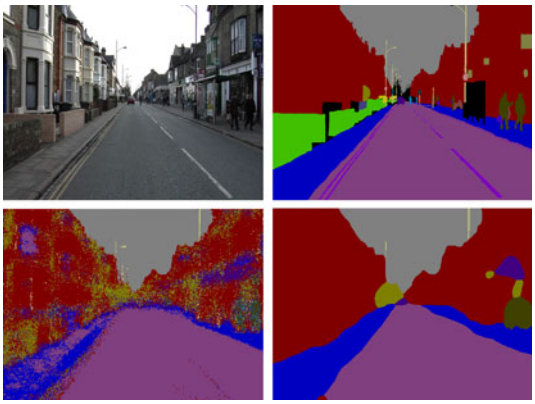

Fig. 2. Examples of object class segmentations using unary classifiers. In row-wise order: original image, ground-truth labelling, random forest, our method.

Once a random forest $\mathcal{F}$ has been trained, the classification of a test image can be naively obtained by considering each pixel to be the centre of a patch and by labelling it with the most probable class predicted by the forest.

3 Learning Structured Class-Labels in Random Forests

In traditional classification approaches like the one presented in the previous section, input data samples are assigned to single, *atomic* class-labels, acting as arbitrary identifiers without any dependencies among them. For many computer vision problems, however, this model is limited because the label space of a classification task does exhibit an inherently topological structure, which renders the class-labels explicitly interdependent. Although this structured label space is already present in the training data, it remains largely unexploited by standard classification approaches, like the random forest introduced in the previous section. Consequently, when applying standard random forest classifiers for semantic image labelling, the obtained results are quite noisy (see e.g., Fig. 2, bottom-left). Indeed, a random patch extracted from the labelled image will likely show a configuration which never appeared in the ground-truth classification used to train the classifiers.

To overcome this limitation, we propose a novel way of enriching the standard random forest classifiers by rendering them aware of the local topological structure of the output label space, as indicated in Fig. 1. To this end, we depart from the traditional classification paradigm and address the problem of learning structured class-labels within the random forest framework.

3.1 Structured Label Space

Our structured label space $\mathcal{P}$ consists of patches of object class-labels. In order to keep the treatment simple we restrict the patch to a specific shape (e.g., square). Hence, we model a patch as a function $p : \mathbb{Z}^2 \rightarrow \mathcal{Y} \cup \{\perp\}$ providing every pixel with a class-label in $\mathcal{Y}$, or with $\perp$ in case no label is assigned. With $p(\mathbf{d})$ we denote the entry of the label patch $p \in \mathcal{P}$ located at $\mathbf{d} \in \mathbb{Z}^2$ and we denote with $\mathrm{dom}(p)$ the set of pixel positions where p holds a label, i.e., $\mathrm{dom}(p) = \{\mathbf{d} \in \mathbb{Z}^2 : p(\mathbf{d}) \in \mathcal{Y}\}$. Additionally, we index the entries in a way that index $(0,0)$ takes the central position. To distinguish between a patch x from the feature space $\mathcal{X}$ (see, Section 2.1) and a patch p from the structured label space $\mathcal{P}$, we refer to them as *feature patch* and *label patch*, respectively. Each training feature patch $x = (\mathbf{u}, I)$ has an associated label patch p which holds the labels of all pixels of image I within the neighbourhood of $\mathbf{u}$ determined implicitly by the patch. Fig. 3 shows an example of a square feature patch x and an associated square label patch centred on pixel $\mathbf{u}$. The label patch holds labels in $\mathcal{Y}$ within the square boundaries and is assumed to be $\perp$ outside the square. Please note that the label patch and the feature patch may have different shapes and dimensions.

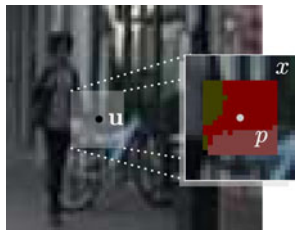

Fig. 3. Training data example for the proposed method. The label for patch x is not limited to the centre label at $\mathbf{u}$, but includes the topology of the local label neighbourhood p.

In the next section we show how the split function selection strategy in the nodes of the random forest will be adapted to account for the new label space. However, for the moment we will simply assume that the training patches from $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{P}$ have been routed through the tree to the leaves. Consider now a leaf t and let $\mathcal{S}_t \subseteq \mathcal{P}$ be the set of structured labels present in the training data used to grow the leaf (see Fig. 4 for some examples of $\mathcal{S}_t$). The class-label π parametrizing the leaf is now a structured label from $\mathcal{P}$ and not just an atomic label from $\mathcal{Y}$ as in the standard random forest. A good selection for the structured class-label should represent a mode of the location-dependent joint distribution $\mathbb{P}[p \,|\, \mathcal{S}_t]$ of the label patches $p \in \mathcal{S}_t$. Accordingly, the label patch π selected for leaf t is the one in $\mathcal{S}_t$ maximizing the joint probability $\mathbb{P}[p \,|\, \mathcal{S}_t]$ (see, Fig. 5):

$$\pi \in \arg\max_{p \in \mathcal{S}_t} \mathbb{P}[p \,|\, \mathcal{S}_t]\,. \tag{4}$$

With a view to keeping the complexity of this step low but simultaneously take into account the topological label

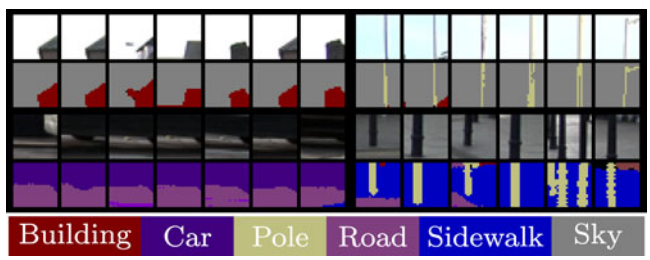

Fig. 4. Illustration of feature patches with corresponding label patches, collected from different leaf nodes when trained on CamVid database. Bottom: Label sets and associated colours.

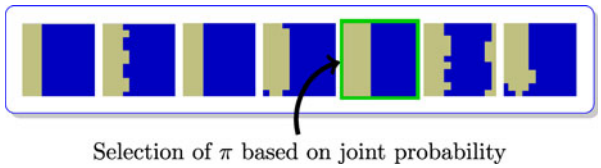

Fig. 5. Example of label patches reaching a leaf during training with the selected prototype.

statistics at absolute positions in the label patches, we empirically estimate the joint probability by making a position-dependent pixel independence assumption as

$$\mathbb{P}[p \mid \mathcal{S}_t] \approx \prod_{\mathbf{d} \in \mathrm{dom(p)}} \mathbb{P}_{\mathbf{d}}[p(\mathbf{d}) \mid \mathcal{S}_t], \tag{5}$$

where $\mathbb{P}_{\mathbf{d}}[y \mid \mathcal{S}_t] = \frac{1}{|\mathcal{S}_t|} \sum_{p \in \mathcal{S}_t} \mathbb{1}[y = p(\mathbf{d})]$ represents the empirical marginal class distribution over all the label patches in $\mathcal{S}_t$ of labels located in position $\mathbf{d}$.

The trivial factorization assumption in (5) does not hold in general and more structured factorizations could be employed (e.g., using Markov random fields). However, we do not require precise estimates at this point since our aim is to detect a good prototype and not to synthesize new label patches. Additionally, our trees tend to cluster together similar label patches (see, e.g., Fig. 4) thus simplifying the complexity of the underlying true joint distribution. From this perspective our proposed, trivial factorization is a good compromise that minimizes the computational overhead without sacrificing the quality of the prototype selection.

3.2 Test Function Selection for Structured Labels

The change introduced in the label space should be coupled with an adaptation of the way a test function is selected in each node of the random forest during the training procedure in order to account for the additional information carried by the structured labels. One naive solution is to port the test selection criterion actually used in the standard random forest to our context, e.g., by simply associating each patch with the label we find in the centre of the associated label patch p. This, however, results in a split of the training set which is identical to what the standard random forest implementation does, without properly exploiting the label topology.

In order to take advantage of the new label space, we propose to select the best split function at each node based on the information gain with respect to a k-label joint distribution. Specifically, we associate each training pair (x, p) with k labels that have been uniformly drawn (once per node) from the patch p. By adopting this new test function selection criterion, all entries of a label patch have the chance to actively influence the way a feature patch is branched through the tree during the training procedure. Of course, one drawback of this new test selection method is the increased complexity deriving from the evaluation of the k-label joint distribution ($|\mathcal{Y}|^k$ elements) instead of the simple, single label distribution ($|\mathcal{Y}|$ elements). Note, however, that if we consider the special case $k = 1$, which consists in associating each training pair (x, p) with just one label, we still have the effect that all entries of the label patch influence the learning procedure, but at no higher computational cost. This is due to the fact that we consider a label $p(\mathbf{d})$ from a random position $\mathbf{d}$ (generated once per node), instead of considering only the central label pixel.

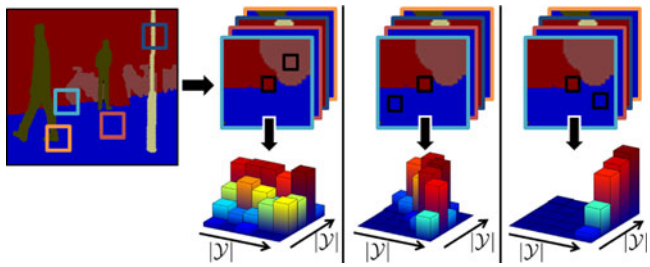

Fig. 6. Left: ground truth label image with training sample locations. Right: schematic illustrations of resulting $k = 2$ joint label distributions evaluated on split candidate sets for different label positions.

In Fig. 6 we provide an illustration when considering the case $k = 2$, i.e., we fix the centre label in the label patches and randomly select a second one. In the training process, the currently investigated split parameters determine the candidate sets for the child nodes where both selected label positions generate the joint label distribution which is used for quantifying the quality of the split. Therefore, using the proposed structured class-labels allows to additionally consider contextual constraints from the label space topology.

3.3 Structured Label Predictions

The structured predictions gathered from the trees of a forest have to be combined into a single label patch prediction. Moreover, predictions obtained by the forest for each image pixel have to be fused together into a consistent labelling. We will address this problem in this section.

In order to determine a single label patch prediction from a set of predictions provided by the trees of a forest, we follow a procedure which is similar to the one adopted in order to select the label patch π in a leaf (see Section 3.1). Consider a trained forest $\mathcal{F}$, a test patch $x = (\mathbf{u}, I)$, and let $\mathcal{H}_x$ be the set of predictions for x gathered from each tree $t \in \mathcal{F}$ (see, Fig. 7) given by

$$\mathcal{H}_x = \{h(x \mid t) \in \mathcal{P} \,:\, t \in \mathcal{F}\}. \tag{6}$$

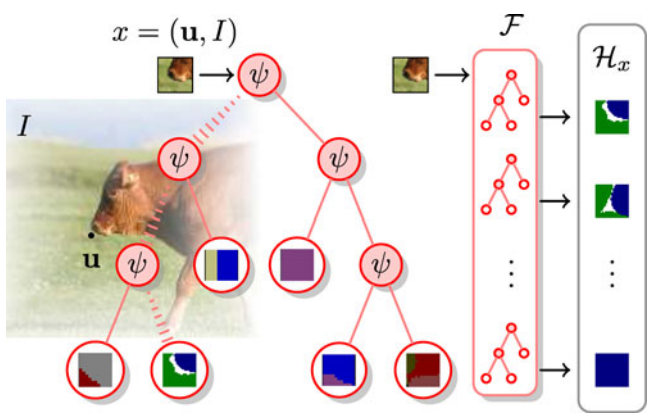

Fig. 7. Prediction pipeline. A sample x is routed through each tree in the forest $\mathcal{F}$ to a leaf where a label patch is assigned. The set of such label patches is given by $\mathcal{H}_x$.

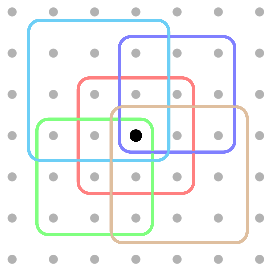

Fig. 8. Simple fusion of structured predictions using 3×3 label patches. Each pixel fuses the class hypotheses gathered from the label patches of neighbouring pixels. For clarity reasons, only 5/9 label patches are drawn.

Like in (4), the label patch prediction of the forest $\mathcal{F}$ for feature patch x is given by the one maximizing the label patch joint probability estimated from $\mathcal{H}_x$, i.e.,

$$p^* \in \arg\max_{p \in \mathcal{H}_x} \mathbb{P}[p \,|\, \mathcal{H}_x], \tag{7}$$

where $\mathbb{P}[p \,|\, \mathcal{H}_x]$ is like (5) with $\mathcal{S}_t$ replaced by $\mathcal{H}_x$.

Unlike standard classification algorithms which, given a test image I, directly assign an object class-label to each pixel, our classifiers cast a prediction for each pixel, involving also the neighbouring ones. Indeed, if $p \in \mathcal{P}$ is the patch label predicted for pixel $\mathbf{u}$ in a test image then a pixel $\mathbf{v}$ in a neighbourhood of $\mathbf{u}$ could be classified as $p(\mathbf{v} - \mathbf{u}) \in \mathcal{Y}$. Hence, for each test pixel we collect a set of class predictions cast from the neighbouring pixels, which have to be integrated into a single class prediction. This process is illustrated in Fig. 8. As we can see, assuming 3×3 square-shaped label patches, each test pixel receives nine class predictions from the neighbourhood, which have to be integrated into a single class prediction. A simple way of performing this integration of votes consists in selecting the most voted class per pixel. We refer to this operation as a *simple fusion*. The outcome of this fusion step is a labelling $\ell = \{\ell_{\mathbf{u}}\}_{\mathbf{u} \in \mathrm{dom(I)}}$ from the set $\mathcal{L}$ of all possible labellings for the image, $\ell_{\mathbf{u}} \in \mathcal{Y}$ being the class-label associated with pixel $\mathbf{u}$.

A different and more principled approach to the computation of the final labelling can be obtained by optimizing the label patch selection with respect to a given labelling rather than solely taking (7) for each pixel. This allows to better exploit the label patch diversity in the set of predictions obtained from (6).

We define the *agreement* of an individual label patch p centred on pixel $\mathbf{v} \in \mathrm{dom}(I)$ in image I with a given labelling $\ell \in \mathcal{L}$ as the number of corresponding pixels sharing the same label, i.e.,

$$\phi^{(\mathbf{v})}(p, \ell) = \sum_{\mathbf{u} \in \mathrm{dom}(I)} \mathbb{1}\left[p(\mathbf{u} - \mathbf{v}) = \ell_{\mathbf{u}}\right]. \tag{8}$$

Furthermore, let $z = \{z_{\mathbf{v}}\}_{\mathbf{v} \in \mathrm{dom(I)}} \in \mathcal{Z}$ be an assignment of label patches to pixels in I, $z_{\mathbf{v}} \in \mathcal{H}_x$ being a label patch for sample $x = (\mathbf{v}, I)$ taken from (6), and $\mathcal{Z}$ denoting the set of all such assignments for image I. For a particular configuration $z \in \mathcal{Z}$ and a labelling $\ell \in \mathcal{L}$, the total agreement $\Phi(z, \ell)$ is defined as the sum of agreements of each label patch in z with the labelling ℓ according to

$$\Phi(z, \ell) = \sum_{\mathbf{v} \in \mathrm{dom(I)}} \phi^{(\mathbf{v})}(z_{\mathbf{v}}, \ell). \tag{9}$$

As we want to find the label patch configuration that leads to the maximum total agreement with the labelling of a test image I, we can write the optimal solution as a pair $(z^*, \ell^*) \in \mathcal{Z} \times \mathcal{L}$, where

$$(z^*, \ell^*) \in \arg\max_{(z,\ell) \in \mathcal{Z} \times \mathcal{L}} \Phi(z, \ell). \tag{10}$$

From a graphical model perspective, (10) coincides with the most probable state of an MRF with a bipartite structure. The bipartite graph consists of two sets of random variables $\ell = \{\ell_{\mathbf{u}}\}_{\mathbf{u} \in \mathrm{dom(I)}} \in \mathcal{L}$ and $z = \{z_{\mathbf{v}}\}_{\mathbf{v} \in \mathrm{dom(I)}} \in \mathcal{Z}$. No edge exists among variables from the same set, while each label variable $\ell_{\mathbf{u}}$ is connected to a label patch variable $z_{\mathbf{v}}$ with an associated energy term given by $E_{\mathbf{uv}}(\ell_{\mathbf{u}}, z_{\mathbf{v}}) = -\mathbb{1}[\mathbb{z}_{\mathbf{v}}(\mathbf{u} - \mathbf{v}) = \ell_{\mathbf{u}}]$. It is straightforward to see that the MRF joint density under this setting is $\mathbb{P}(z, \ell) \propto \exp(\Phi(z, \ell))$ and, hence, that the most likely state, i.e., the one maximizing $\log \mathbb{P}(z, \ell)$, is a solution of (10). Note that the dependence of the graphical model on the test image is determined by shaping the values that the label patch variables can take. Alternatively, one could also obtain an interpretation in terms of a CRF, by assuming a label patch variable $z_{\mathbf{u}}$ to take any value in $\mathcal{P}$ and the corresponding data term to inflict an infinite cost on the patches that are not provided by our forest for pixel $\mathbf{u}$, and a zero cost on all other, admissible patches.

The optimization problem in (10) is in general non-trivial to solve. The algorithm we propose is a heuristic, which is simple and effective as shown by our experiments. It is based on an alternating optimization technique, where we iteratively switch between optimizing the labelling variable $\ell \in \mathcal{L}$ given a configuration of label patches (we apply a simple fusion step) and optimizing the configuration variable $z \in \mathcal{Z}$ given a labelling for image I. This algorithm can be regarded as a special instance of the block-wise iterated conditional modes method [55], [56] considering the MRF interpretation given above, where we alternate between optimizing the block of label variables and the block of label patch variables. Note that this choice for the blocks is computationally appealing as the label variables are conditionally independent, given the label patch assignments and vice-versa. Details about the algorithm can be found in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2014.2315814.

4 Enhancing Object Detection with Structured Class-Labels

In the computer vision literature the random forest framework has also been successfully applied to object detection, i.e., the problem of localizing multiple instances of a specific object class within an image. In [14], [57], a variant of the random forest known as *Hough forest* has been introduced which allows to implement a generalized Hough transform [58] together with a per-sample classification. In this sense, Hough trees combine properties of classification and

regression trees and are able to perform mixed, discrete- and continuous-valued predictions.

In this section, we propose a modification over the standard Hough forest framework [57] in order to cope with label patches instead of atomic class-labels, in the presence of pixel-wise ground-truth annotations. The substantial difference with respect to the original Hough forest solution lies in the training phase, where we employ our split function selection criterion for structured labels (as described in Section 3) to shape the tree in a structure-aware flavour. Exploiting the additional information delivered by the label patches yields to a considerable, positive impact on the generalization capabilities of the forest and allows us to achieve state-of-the-art performance on pedestrian detection tasks as we will show in the experimental section.

The basic idea behind the generalized Hough transform [58], which extends the original work of Hough [59], is to abstract the input image into voting elements and conduct the detection process in a voting space, the so-called Hough space. Each voting element is allowed to cast a directional vote for a specific location in the Hough space. The entries in these locations are grouped in a point-wise, accumulative style and are associated to certain hypotheses, indicating the confidences for object presence. The quality of a hypothesis, i.e., its certainty about object presence, depends on the associated peak in the Hough domain. Hough trees learn a mapping between the appearance of an image patch and its relative position to the object category centroid (i.e., centre voting information). During inference, the forest allows to perform both classification on test samples and cast probabilistic votes in a generalized Hough-voting space that is subsequently analysed to obtain object centre hypotheses. In particular, the final detection is achieved by applying non-maximum suppression (NMS) in the voting space to detect the strongest hypotheses.

More formally, the input space $\mathcal{X}$ for the Hough forest is a set of patches. Each feature patch is represented as a pair $(\mathbf{u}, I)$, where $\mathbf{u} \in \mathbb{Z}^2$ is the patch centre in image $I \in \mathcal{I}$. The output space $\mathcal{Y}$ is a set of pairs $(p, \mathbf{d})$, where $p \in \mathcal{P}$ is a binary label patch, its classes indicating the presence of an object and $\mathbf{d} \in \mathbb{Z}^2$ is a displacement vector to the object's centroid. In such a way, each training sample $(\mathbf{u}, I) \in \mathcal{X}$ is equipped with ground truth information $(p, \mathbf{d}) \in \mathcal{Y}$. We say that a sample is a *background* sample if the central pixel of the label patch p has class BG; otherwise, we say that it is a *foreground* sample and has class FG. The information about the displacement vector is ignored in case of background samples, while it indicates the object centroid at location $\mathbf{u} + \mathbf{d}$ in image I in case of foreground samples.

Following [57], instead of storing an element of $\mathcal{Y}$ in each leaf of the forest, we keep a joint probability distribution q for the background/foreground label in $\{\texttt{BG}, \texttt{FG}\}$ and the displacement vector in $\mathbb{Z}^2$ to an object's centroid, i.e., $q \in \mathbb{P}(\{\texttt{BG}, \texttt{FG}\} \times \mathbb{Z}^2)$, where the notation $\mathbb{P}(\mathcal{Y})$ refers to the set of probability distributions having $\mathcal{Y}$ as sample space. By doing so, we have more flexibility in the subsequent voting process. The distribution provided in each leaf factorizes into two marginal distributions, for the class-labels and the displacement vectors, respectively. The marginal over the class-labels is a discrete, binary distribution $q^{\text{class}} \in \mathbb{P}(\{\texttt{BG}, \texttt{FG}\})$ providing the probability of drawing a background or a foreground sample from the set $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ of training samples reaching the leaf: $q^{\text{class}}(c|\mathcal{D}) = |\mathcal{D}_c|/|\mathcal{D}|$, where $\mathcal{D}_{\texttt{BG}}$ and $\mathcal{D}_{\texttt{FG}}$ are subsets of training samples in $\mathcal{D}$, with a label patch having the central pixel labelled as FG and BG, respectively. The marginal probability density over the displacement vectors, denoted as $q^{\text{vote}} \in \mathbb{P}(\mathbb{Z}^2)$, is defined using a Parzen window with a Gaussian kernel function over the set of displacement vectors collected from the foreground training samples: $q^{\text{vote}}(\mathbf{d} \,|\, \mathcal{D}) \propto \sum_{\mathbf{d}'} K_\sigma(\|\mathbf{d} - \mathbf{d}'\|)$, where K_σ is a Gaussian kernel with bandwidth σ and $\mathbf{d}'$ runs over all displacement vectors of samples in $\mathcal{D}_{\texttt{FG}}$. Consequently, a Hough forest stores in each leaf a probability distribution $q(c, \mathbf{d} \,|\, \mathcal{D}) = q^{\text{class}}(c \,|\, \mathcal{D}) \cdot q^{\text{vote}}(\mathbf{d} \,|\, \mathcal{D})$.

Inference. Given a new test sample $x = (\mathbf{u}, I) \in \mathcal{X}$, a Hough tree $t \in \mathcal{F}$ from the forest $\mathcal{F}$ provides the posterior probability for a possible object centre hypothesis $E(\mathbf{v})$ in position $\mathbf{v}$ as

$$\mathbb{P}(E(\mathbf{v}) \,|\, x, t) = q(\texttt{FG}, \mathbf{v} - \mathbf{u}), \tag{11}$$

where q is the distribution stored in the leaf of the Hough tree t reached by the sample x. The posterior probability given the whole forest $\mathcal{F}$ is obtained as

$$\mathbb{P}(E(\mathbf{v}) \,|\, x, \mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{t \in \mathcal{F}} \mathbb{P}(E(\mathbf{v}) \,|\, x, t)\,. \tag{12}$$

Training. The training procedure for Hough trees aims at reducing the prediction error on the training set and generalizing to unseen test data. The key difference to the training procedure adopted for standard decision trees is that the training objective minimizes both the *class-label uncertainty* and the *offset uncertainty* based on a randomized decision. The class-label uncertainty of a set of samples $\mathcal{D}$, denoted by $U^{\text{class}}(\mathcal{D})$, is given by the weighted entropy measure over its class-label distribution:

$$U^{\text{class}}(\mathcal{D}) = |\mathcal{D}|\, \mathbb{H}(\mathcal{D})\,. \tag{13}$$

Differently from [57] that uses atomic foreground/background class labels, and in order to exploit the additional information delivered by the label patches, we consider here a joint probability distribution over the label patches in the computation of the entropies as described in Section 3.2. On the other side, the offset uncertainty of a set of samples $\mathcal{D}$, denoted by $U^{\text{vote}}(\mathcal{D})$, is the non-normalized variance in the set of centroid voting vectors from foreground samples in $\mathcal{D}$. This can be formalized as

$$U^{\text{vote}}(\mathcal{D}) = \sum_{\mathbf{d}} \left\|\mathbf{d} - \overline{\mathbf{d}}\right\|^2, \tag{14}$$

where $\overline{\mathbf{d}} = \sum_{\mathbf{d}} \mathbf{d}/|\mathcal{D}_{\texttt{FG}}|$ is the mean voting vector of the foreground samples reaching the node. In both equations we assume that $\mathbf{d}$ runs over all displacement vectors of samples in $\mathcal{D}_{\texttt{FG}}$. Accordingly, $U^{\text{vote}}(\cdot)$ considers only foreground training samples, while all background samples are ignored, because only the foreground samples contribute with votes in the Hough space. As a result, the training objective function seeks for the optimal parameters

TABLE 1
Classification Results on CamVid Database for Label Patch Size 13×13 and Comparisons to Related Works

Method	*Global*	*Avg(Class)*	*Avg(Pascal)*
RF with Motion + Structure [60]	61.8	43.6	-
[60] + Appearance features	69.1	53.0	-
Local label descriptors [39]	73.7	36.3	29.6
Baseline RF in [47]	-	-	33.3
Baseline RF+CRF [47]	-	-	41.7
Best performing GeoF [47]	-	-	38.3
Our Baseline RF	69.9	42.2	30.6
Our Baseline RF + CRF	74.5	45.4	33.8
Structure + Simple Fusion	74.8	45.0	34.1
2-Full + Simple Fusion	76.8	46.1	35.4
2-Full + Optimized Selection	79.2	46.0	36.2
2-Full + Optim. Sel. + Corr. coeff.	83.8	53.2	43.5
3-Full + Optim. Sel. + Corr. coeff.	82.0	52.1	41.6

$$\psi \in \arg\min_{\psi' \in \Psi(\mathcal{D})} \left\{ \sum_{i \in \{l,r\}} U^{\star}(\mathcal{D}_i^{\psi'}) \right\}, \tag{15}$$

where $\star = \{\text{class}, \text{vote}\}$ randomly selects (according to a uniform distribution) either the class-label uncertainty U^{class} or the offset uncertainty U^{vote}.

5 Experiments

In this section we use our algorithm for the task of semantic segmentation on the CamVid [60] and MSRCv2 [61] databases, for the task of inpainting on the KAIST Hanja2 database and for the task of object detection on two TU Darmstadt pedestrian databases. For performance reasons, we implemented our method in C++ and ran all experiments on a standard desktop computer with 2.9 GHz and 2 GB RAM.

In all our experiments we show a comparison to a standard random forest implementation (denoted as "Our Baseline RF"), which is actually a special instance of our method with a centred label patch size of 1×1. We report scores of related results where applicable and additionally provide a comparison to a CRF-based approach (using the output of the baseline RF as unary and a contrast-sensitive Potts model as pairwise term, respectively). To show the impact of the respective stages of our method, we evaluate different training ["Structure" / "k-Full"] and classification ["Simple Fusion" / "Optimized Selection"] procedures as follows: "Structure" uses the structured label patches but only considers the label distribution at a single, randomly selected label position for training. "k-Full" considers structured label patches and k-label joint distributions in the split functions (see Section 3.2 for more details). "Simple Fusion" and "Optimized Selection" refer to the fusion methods of the structured output predictions as described in Section 3.3.

We used the same low-level image features for training the baseline and our novel structured learning random forests: CIELab raw channel intensities, first and second order derivatives as well as HOG-like features, computed on the L-channel. Moreover, we show results when additionally using correlation coefficients between covariances of the RGB raw channel intensities and the first order derivatives of the grayscale intensity image, similar to [62], [63]. In all experiments on semantic segmentation we fixed the feature patch size to 24×24 and trained 15 trees, using 500 iterations for the node tests and stopping when less then 10 samples per leaf were available.

TABLE 2
Classification Results on CamVid Database as a Function of the Label Patch Size Using Simple Fusion

Score	1×1	3×3	5×5	7×7	9×9	11×11	13×13
Global	70.0	73.7	75.1	75.6	76.1	76.5	76.9
Avg(class)	42.2	45.3	46.4	46.4	46.1	45.9	46.0
Avg(Pascal)	30.6	33.6	34.8	35.0	35.1	35.3	35.4

We list the scores of our experiments according to the same evaluation criteria used in [60], [61], [63] and additionally include the more strict average intersection vs. union score as e.g., used in the PASCAL VOC challenges [64]. In particular, "*Global*" refers to the percentage of all pixels that were correctly classified, "*Avg(Class)*"[2] expresses the average recall over all classes and "*Avg(Pascal)*"[3] denotes the average intersection versus union (or Jaccard) score.

5.1 CamVid Database Experiments

The Cambridge-driving Labelled Video Database [60] is a collection of videos captured on road driving scenes. It consists of more than 10 minutes of high quality (970×720), 30 Hz footage and is divided into four sequences. Three sequences were taken during daylight and one at dusk. A subset of 711 images is almost entirely annotated into 32 categories, but we used only the 11 commonly used categories with the same splits for training and testing as presented in [60], [65].

We resized the images by a factor of 0.5 and randomly collected training samples on a regular lattice with a stride of 10, resulting in approximately 850k training samples. The training time per tree is 23 minutes when using the single label test and 30 minutes with the joint label test $k = 2$. For the experiment where we only consider the labelling transitions, we reduced the stride to 8. In order to correct the imbalance among samples of different classes, we applied an inverse frequency weighting as mentioned in Section 2.

CamVid - 11 classes. The standard protocol for evaluating on the CamVid database considers the following 11 object categories, forming a majority of the overall labelled pixels (89.16 percent): Road, Building, Sky, Tree, Sidewalk, Car, Column_Pole, Sign-Symbol, Fence, Pedestrian and Bicyclist. In Table 1 we list our results using a label patch size of 13×13, clearly indicating the performance boost when using our learning method over the standard random forest. Both, Simple Fusion and Optimized Selection strategies improve over the baseline forests and the best-performing method is approximately on par with the RF+CRF experiment in [47] where a more complex pairwise model was used as the one in (Our Baseline RF + CRF).

In Table 2 we show the influence of the label patch size, i.e., the size of the considered label topology during training and classification using the configuration '2-Full + Simple

2. (True Pos.)/(True Pos. + False Neg.)
3. (True Pos.)/(True Pos. + False Neg. + False Pos.)

TABLE 3
Classification Results for Labelling Transitions on the CamVid Database for Label Patch Size 11×11

Method	*Global*	*Avg(Class)*	*Avg(Pascal)*
Our Baseline RF	63.8	44.2	29.8
Our Baseline RF + CRF	68.2	48.2	33.3
Structure + Simple Fusion	69.9	50.4	35.0
2-Full + Simple Fusion	71.6	50.1	35.8
2-Full + Optimized Selection	72.5	51.4	36.4

Fusion'. It can be seen that even a small neighbourhood ($\geq 5 \times 5$) leads to a significant boost in the classification stage.

Labelling transition evaluation. In this experiment we evaluate only the transitions between object classes to demonstrate the impact of structured predictions on the label border classification results. To perform this experiment, we discard all labels in the ground truth information lying outside a radius of 24 pixels to a transition between two or more classes. In Table 3, the corresponding results are listed when using a label neighbourhood of 11×11. Although the global score has slightly dropped compared to the previous experiment, we obtain improvements on the (stricter) *Avg(Class)* and *Avg(Pascal)* criteria. This strengthens the claim that our framework yields superior results, especially when classifying local label transitions of object classes.

5.2 MSRCv2 Database Experiments

To show that our method also yields an improvement when the images are not entirely labelled, we performed another experiment on the MSRCv2 Database [61]. This database consists of 532 images containing 21 object classes and predefined splits into 276 training and 256 test images. We collected the training samples on a regular lattice with a stride of 5, leading to approximately 500k training samples and training times of 13 and 17 minutes per tree using single or joint label distributions, respectively. In Fig. 9 we show some qualitative results and in Table 4, we provide the scores for a label neighbourhood size of 11×11 and again find an improvement with our structured learning algorithm. The gain of using the joint statistics over the single label distribution seems to vanish in the Simple Fusion approach, however, we explain this by the fact that our algorithm does not see enough properly labelled transitions between different classes.

TABLE 4
Classification Results on MSRCv2 Database for Label Patch Size 11×11

Method	*Global*	*Avg(Class)*	*Avg(Pascal)*
Texton forests naïve (superv.) [44]	49.7	34.5	-
Texton forests (Full system) [44]	72.0	67.0	-
RF using covariance features [63]	55.8	42.2	-
Our Baseline RF	54.8	43.4	28.3
Our Baseline RF + CRF	61.0	52.8	35.1
Structure + Simple Fusion	60.8	51.0	33.8
2-Full + Simple Fusion	60.8	51.1	33.9
2-Full + Optimized Selection	63.9	55.6	37.6
2-Full + Optim. Sel. + Corr. coeff.	70.0	59.6	43.3

5.3 KAIST Hanja2 Database

Here we demonstrate that our method can also be used for reconstructing occluded regions in handwritten, Chinese characters of the KAIST Hanja2 Database. We reproduced an experiment from the work in [35], aiming to learn calligraphy properties. In their work, the authors proposed a method for learning class-specific, distant contextual models by combining random decision trees and random fields in a so-called *decision tree field*. The purpose of this experiment is to show that inpainting results can be improved with structured class-labels, since they exhibit local shape information in the binary classification case.

We used the original training (300 images) and testing data (100 images) of [35] and their respective, randomly generated occlusions for the *small* occlusion database. As a baseline, we obtain an average per-tree, pixel-wise classification result of 68.52 percent when evaluating 10 randomly trained decision trees (with a maximum depth of 15). We used 2,000 iterations per node training and simple pixel difference tests on the intensity values, which were allowed to look at most 80 pixels away. We used the identical setup within our structured class-label random trees, considering label patches of size 5×5.

In Table 5, we compare the classification results when using only our single, baseline decision tree (DT), a tree ensemble of 10 trees (RF), the Markov Random Field, the Decision Tree Field (DTF), the Regression Tree Field (RTF) approaches (using regression trees with maximum depth 20 and results taken from [35], [36]) and the convex quadratic relaxation approach (QP_{M3N}) of [66]. Additionally, we compare to our recent work in [37] as described in the related work section. With our proposed structured class-labels, we can boost the initial classification score of the single decision tree by almost 10 percent, outperforming sophisticated

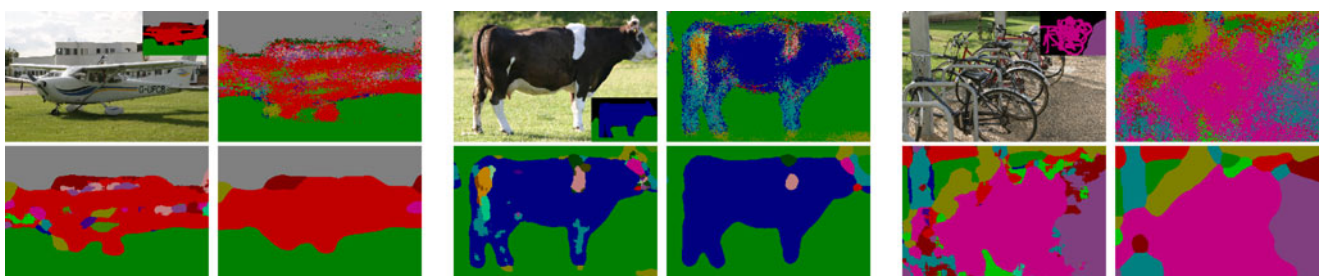

Fig. 9. Qualitative results from the MSRCv2 database. (top-left) Original image with ground truth annotation, (top-right) standard random forest, (bottom-left) 2-Full + Simple Fusion, (bottom-right) 2-Full + Optimized Selection.

TABLE 5
Reconstruction Results for KAIST Hanja2 Small Occlusions Database in [Percent]

DT (Avg)	RF	MRF [35]	DTF [35]	RTF_{1D} [36]	RTF_{2D} [36]	SLP [37]	QP_{M3N} [66]	Ours (5×5, k=2)
68.52	74.95	75.18	76.01	76.39	77.55	78.07	**79.36**	78.09

methods like MRF, DTF or RTF. The result is approximately on par with SLP and slightly below QP_{M3N}, however, without the need for explicit parametrization of the pairwise interactions to be learned or solving an optimization problem during inference, respectively. Following the works in [35], [66], we also illustrate qualitative results on the large occlusion database (see Fig. 10).

5.4 Person Detection on TU Darmstadt Databases

To evaluate our approach on object detection problems as described in Section 4, we have conducted experiments on the task of pedestrian detection. Detecting pedestrians is very challenging for Hough-voting based methods alone, as they typically exhibit strong articulations of feet and arms, leading to non-distinctive hypotheses in the Hough space. However, with our combined approach we also learn the local shape information by analysing the joint label statistics of the structured class-labels from the (binary) ground truth labelling, expecting the trees to better capture the respective locations of the object parts.

We evaluated our method on the TUD pedestrian databases [67], showing our detection results with training according to the standard protocol using 400 training images (where each image contains a single annotation of a pedestrian) and evaluation on the *Campus* and *Crossing* scenes, respectively. For evaluation on the Crossing scene, we used the annotations from [68], providing a total number of 1,216 bounding boxes. This annotation is even more detailed than the one presented in [69] with 1,018 bounding boxes. For our experiments, we rescaled the images by a factor of 0.5 and doubled the training image set by including also the horizontally flipped images. We randomly chose 125 training samples per image for foreground and background, resulting in $2 \cdot 400 \cdot 2 \cdot 125 = 200\,\mathrm{k}$ training samples per tree.

As a baseline, we use the Hough Forest [57] with the same training parameters used for our enhanced, structured class-label Hough forest. We trained 20 trees and the training data was sampled homogeneously per category per image. The patch size was fixed to 20×20, the joint-label distribution parameter was fixed to $k = 2$ and we performed 1,600 node tests to find the best split function parameters per node. The trees were stopped growing when < 7 samples were available. As image features, we used the first 16 feature channels provided in the publicly available Hough Forest code of [57]. In order to obtain the object detection hypotheses from the Hough space, we used the same Non-maximum suppression technique in all our experiments as suggested in [57]. To evaluate the obtained hypotheses, we use the standard PASAL-VOC criterion which requires the mutual overlap between ground truth and detected bounding boxes to be ≥ 50 percent.

For additional comparisons we report the results of [68], from where we also provide evaluation results of the Implicit Shape Model (ISM) [70]. Please note that the results of [68] are based on a different baseline implementation. Additionally, we include the results obtained with the publicly available code for the approach of [69]. Finally, we also list the scores obtained in our recent work [46] being trained according to the same training protocol and parameters as for our structured class-label Hough forest.

5.5 Evaluation on Campus Scene

First, we discuss the results obtained on the Campus scene. This database consists of 71 images showing walking pedestrians at severe scale differences and partial occlusions. The ground truth we use has been released with [69] and contains a total number of 314 pedestrians.

Fig. 11 shows precision/recall curves when evaluating on three scales (factors 0.3, 0.4, 0.55) on the left and using

Fig. 10. Reconstructions of Chinese characters (large occlusions). Left: ground truth image. Middle: occluded test image. Right: our reconstruction.

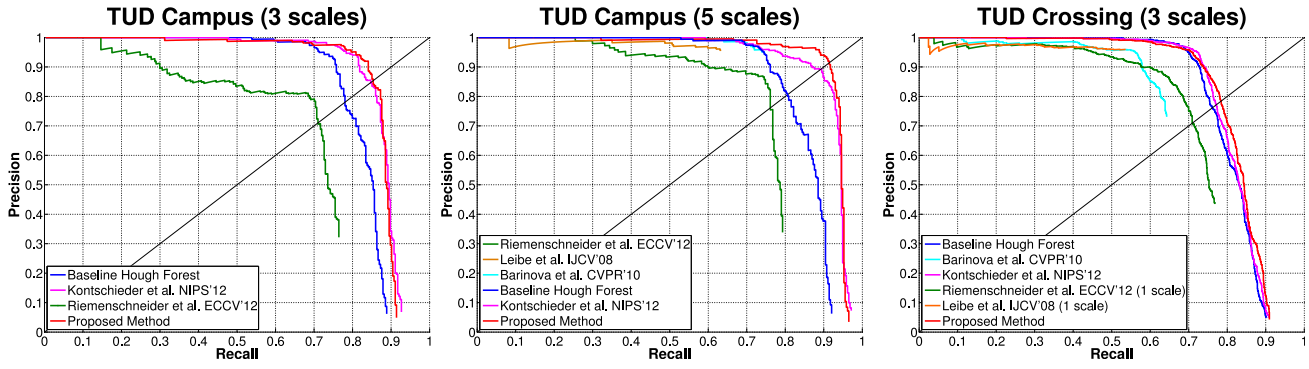

Fig. 11. Precision-Recall curves for pedestrian detection experiments on TUD Campus scene (left & middle plots) and Crossing scene (right plot).

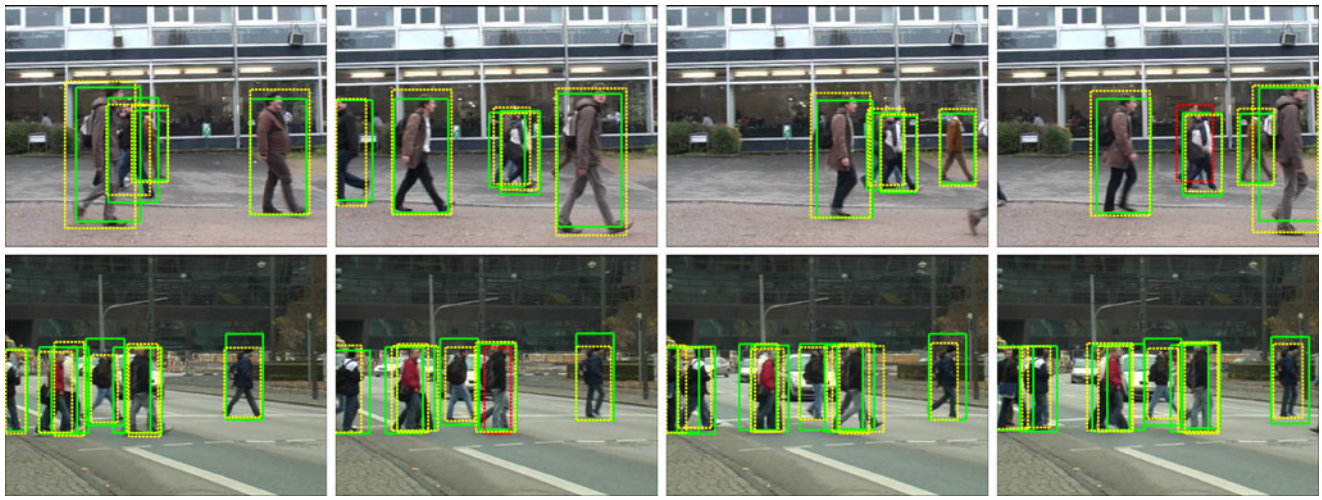

Fig. 12. Detection results on Campus scene (five scales) on top row and Crossing scene (three scales) on bottom row. Bounding box colours: Yellow for ground truth, green for obtained detections and red for missed detections.

five scale factors (0.34, 0.42, 0.51, 0.65, 0.76) in the middle. When considering three scales and a precision rate of 90 percent, we obtain a considerable improvement of ≈ 8 percent over the Hough forest baseline implementation. Moreover, we also outperform our recent method [46] by ≈ 2 percent. When using five scales we get even better results, i.e., we improve by ≈ 16 percent over the baseline Hough forest and still ≈ 5 percent over our earlier work [46]. We believe that the improvement over the standard Hough forest is due to better voting vector separation abilities of foreground data in the split nodes. In addition to using relative positions of the training samples to the object centroid, also the local shape (e.g., human silhouette) encoded in the label information is exploited in the training process. This hypothesis might also be supported by the fact that five scales yield higher detection rates than three scales. In fact, appearance and scale should of course be as close as possible to the ones observed during learning. Even though the improvement over [46] is smaller, we stress that in the proposed approach we do not explicitly learn from nearby object entities as it is possible in [46]. However, such object-instance specific information might also be included here.

Fig. 12 shows exemplary results of our method. Yellow dashed bounding boxes indicate ground truth, green solid boxes are our detected hypotheses and red are missed detections.

5.6 Evaluation on Crossing Scene

The right plot in Fig. 11 shows the results when the same forests are tested on the Crossing scene, using the ground truth annotations of [68]. This database shows walking pedestrians (see Fig. 12, bottom row) with a smaller variation in scale compared to the Campus scene but with strong mutual occlusions and overlaps. We still find an improvement with respect to the baseline (≈ 2 percent gain at a precision of 90 percent) and are about on par with our earlier work in [46]. Since the training data only contains single object instances, i.e., each training image contains exactly one annotated person, mutual occlusions cannot directly be learned from this data. The illustrations in Fig. 12 show how we are able to accurately outline the respective object locations, even in severely occluded areas. The provided failure case illustrates that the annotations are very strict and contain also highly occluded objects, making it a very challenging database.

6 Conclusions

In this paper we presented a simple and effective way to integrate ideas from structured learning into the popular random forest framework for the task of semantic image labelling and object detection. In particular, we incorporated the topology of the local label neighbourhood in the training process and therefore intuitively learned valid labelling transitions among adjacent object categories. During the tree construction, we used topological joint label statistics of the training data in the node split functions for exploring the structured label space. For classification, we provided two possible options for fusing the structured label predictions: A simple method using overlapping predictions and a more principled approach, selecting most compatible label patches in the neighbourhood. Moreover, we have integrated the concept of Hough voting for the task of object detection, showing how structured labels support the learning process for offset regression. We provided several experiments for both, semantic segmentation and object detection and found superior results when compared to standard random forest and conditional random field (using pairwise potentials) classification results.

References

[1] S. Cohen, "Knowledge and context," *The J. Philosophy*, vol. 83, no. 10, pp. 574–583, 1986.
[2] A. W. Price, *Contextuality in Practical Reason*. London, U.K.: Oxford Univ. Press, 2008.
[3] C. K. Chow, "A recognition method using neighbor dependence," *IRE Trans. Electron. Comput.*, vol. 11, pp. 683–690, 1962.
[4] G. T. Toussaint, "The use of context in pattern recognition," *Pattern Recognit.*, vol. 10, pp. 189–204, 1978.
[5] A. Rosenfeld, R. Hummel, and S. W. Zucker, "Scene labeling by relaxation operations," *IEEE Trans. Syst., Man Cybern.*, vol. 6, no. 6, pp. 420–433, Jun. 1976.

[6] R. A. Hummel and S. W. Zucker, "On the foundations of relaxation labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, no. 3, pp. 267–287, May 1983.

[7] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[8] J. R. Quinlan, "Induction of decision trees," *Mach. Learning*, vol. 1, pp. 81–106, 1986.

[9] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, 1997.

[10] L. Breiman, "Random forests," *Mach. Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] A. Criminisi and J. Shotton, *Decision Forests in Computer Vision and Medical Image Analysis*. New York, NY, USA: Springer, 2013.

[12] T. Sharp, "Implementing decision trees and forests on a GPU," in *Proc. Eur. Conf. Comput. Vis.*, 2008, vol. 5305, pp. 595–608.

[13] R. Caruana, N. Karampatziakis, and A. Yessenalina, "An empirical evaluation of supervised learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learning*, 2008, pp. 96–103.

[14] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2188–2202, Nov. 2011.

[15] P. Kontschieder, S. Rota Bulò, M. Donoser, M. Pelillo, and H. Bischof, "Semantic image labelling as a label puzzle game," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 111.1–111.12.

[16] P. Kontschieder, S. Rota Bulò, H. Bischof, and M. Pelillo, "Structured class-labels in random forests for semantic image labelling," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2190–2197.

[17] Y. Yakimovsky and J. Feldman, "A semantics-based decision theory region analyzer," in *Proc. 3rd Int. Joint Conf. Artif. Intell.*, 1973, pp. 580–588.

[18] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. 22, no. 1, pp. 67–92, Jan. 1973.

[19] A. Hanson and E. Riseman, "Visions: A computer vision system for interpreting scenes," in *Computer Vision Systems*, New York, NY, USA: Elsevier, 1978, pp. 303–334.

[20] M. Pelillo, "The dynamics of nonlinear relaxation labeling processes," *J. Math. Imag. Vis.*, vol. 7, no. 4, pp. 309–323, 1997.

[21] M. Pelillo and M. Refice, "Learning compatibility coefficients for relaxation labeling processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 9, pp. 933–945, Sep. 1994.

[22] S. Z. Li, *Markov Random Field Modeling in Computer Vision*. London, U.K.: Springer-Verlag, 1995.

[23] B. Russell, A. Torralba, C. Liu, R. Fergus, and W. Freeman, "Object recognition by scene alignment," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1241–1248.

[24] A. Shinghal, J. Luo, and W. Zhu, "Probabilistic spatial context models for scene content understanding," in *Proc. IEEE CS Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. 235–241.

[25] A. Torralba, "Contextual priming for object detection," *Int. J. Comput. Vis.*, vol. 53, no. 2, pp. 153–167, 2003.

[26] P. Carbonetto, N. Freitas, and K. Barnard, "A statistical model for general contextual object recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2004, vol. 3021, pp. 350–362.

[27] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learning*, 2001, pp. 282–289.

[28] X. He, R. S. Zemel, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *Proc. IEEE CS Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 695–703.

[29] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[30] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, 2007.

[31] L. Ladický, C. Russell, P. Kohli, and P. Torr, "Graph cut based inference with co-occurrence statistics," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 239–253.

[32] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr, "What, where and how many? Combining object detectors and CRFs," in *Proc. Eur. Conf. Comput. Vis.*, 2010, vol. 6314, pp. 424–437.

[33] J. M. Gonfaus, X. Boix, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez, "Harmony potentials for joint classification and segmentation," in *Proc. IEEE CS Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3280–3287.

[34] I. Biederman, "Perceiving real-world scenes," *Science*, vol. 177, no. 7, pp. 77–80, 1972.

[35] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli, "Decision tree fields," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1668–1675.

[36] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother, "Regression tree fields—An efficient, non-parametric approach to image labeling problems," in *Proc. IEEE CS Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2376–2383.

[37] S. Rota Bulò, P. Kontschieder, M. Pelillo, and H. Bischof, "Structured local predictors for image labelling," in *Proc. IEEE CS Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3530–3537.

[38] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille, "Recursive segmentation and recognition templates for 2D parsing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1985–1992.

[39] Y. Yang, Z. Li, L. Zhang, C. Murphy, J. Hoeve, and H. Jiang, "Local label descriptor for example based semantic image labeling," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, vol. 7578, pp. 361–375.

[40] S. Nowozin and C. H. Lampert, "Structured learning and prediction in computer vision," *Found. Trends Comput. Graph. Vis.*, vol. 6, no. 3–4, pp. 185–365, 2011.

[41] Z. Tu, "Auto-context and its application to high-level vision tasks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[42] A. Torralba, K. Murphy, and W. Freeman, "Contextual models for object detection using boosted random fields," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, vol. 17, pp. 1401–1408.

[43] S. Avidan, "Spatialboost: Adding spatial reasoning to AdaBoost," in *Proc. Eur. Conf. Comput. Vis.*, 2006, vol. 3954, pp. 386–396.

[44] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[45] A. Montillo, J. Shotton, J. Winn, J. E. Iglesias, D. Metaxas, and A. Criminisi, "Entangled decision forests and their application for semantic segmentation of CT images," in *Proc. 22nd Int. Conf. Inf. Process. Med. Imag.*, 2011, pp. 184–196.

[46] P. Kontschieder, S. Rota Bulò, A. Criminisi, P. Kohli, M. Pelillo, and H. Bischof, "Context-sensitive decision forests for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 440–448.

[47] P. Kontschieder, P. Kohli, J. Shotton, and A. Criminisi, "GeoF: Geodesic forests for learning coupled predictors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 65–72.

[48] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013.

[49] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1577–1584.

[50] A. Bosch, A. Zisserman, and X. Muñoz, "Image classification using random forests and ferns," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[51] C. Leistner, A. Saffari, J. Santner, and H. Bischof, "Semi-supervised random forests," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 506–513.

[52] C. Leistner, A. Saffari, and H. Bischof, "MIForests: Multiple-instance learning with randomized trees," in *Proc. Eur. Conf. Comput. Vis.*, 2010, vol. 6316, pp. 29–42.

[53] V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 775–781.

[54] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 985–992.

[55] J. Kittler and J. Föglein, "Contextual classification of multi-spectral pixel data," *Image Vis. Comput.*, vol. 2, no. 1, pp. 13–29, 1984.

[56] B. M. Kelm, N. Müller, B. H. Menze, and F. A. Hamprect, "Bayesian estimation of smooth parameter maps for dynamic contrast-enhanced MR images with block-ICM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, 2006, pp. 96–103.

[57] J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1022–1029.

[58] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.
[59] P. Hough, "Machine analysis of bubble chamber pictures," in *Proc. Int. Conf. High Energy Accelerators Instrum.*, 1959, pp. 554–558.
[60] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 44–57.
[61] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2006, vol. 3951, pp. 1–15.
[62] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on Lie algebra," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 1, pp. 728–735.
[63] S. Kluckner, T. Mauthner, P. M. Roth, and H. Bischof, "Semantic image classification using consistent regions and individual context," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 25.1–25.12.
[64] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[65] P. Sturgess, K. Alahari, L. Ladický, and P. Torr, "Combining appearance and structure from motion features for road scene understanding," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 62.1–62.11.
[66] J. Jancsary, S. Nowozin, and C. Rother, "Learning convex QP relaxations for structured prediction," in *Proc. Int. Conf. Mach. Learning*, 2013, vol. 28, pp. 915–923.
[67] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[68] H. Riemenschneider, S. Sternig, M. Donoser, P. M. Roth, and H. Bischof, "Hough regions for joining instance localization and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, vol. 7574, pp. 258–271.
[69] O. Barinova, V. Lempitsky, and P. Kohli, "On detection of multiple object instances using hough transforms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2233–2240.
[70] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vis.*, vol. 77, no. 1–3, pp. 259–289, 2008.

Peter Kontschieder received the MSc and PhD degrees from the Graz University of Technology in Austria, in 2008 and 2013, respectively. He is currently a postdoctoral researcher in the Machine Learning and Perception Group at Microsoft Research in Cambridge, United Kingdom. His research interests include ensemble learning methods with focus on random decision forests and their customization for computer vision problems like object detection and semantic segmentation. During his PhD studies he was a visiting researcher at the Ca'Foscari University of Venice (2011) and accomplished an internship at Microsoft Research Cambridge (2012). He published his research in high-impact conferences like ICCV, CVPR, and NIPS.

Samuel Rota Bulò received the PhD degree in computer science from the University of Venice, Italy, in 2009. He worked as a postdoctoral researcher at the same institution until 2013. He is currently a researcher of the "Technologies of Vision" laboratory at Fondazione Bruno Kessler in Trento, Italy. His main research interests include the areas of computer vision and pattern recognition with particular emphasis on discrete and continuous optimisation methods, graph theory, and game theory. His additional research interests include the field of stochastic modelling. He regularly publishes his research in well-recognized conferences and top-level journals mainly in the areas of computer vision and pattern recognition. He held research visiting positions at the following institutions: IST - Technical University of Lisbon, University of Vienna, Graz University of Technology, University of York, United Kingdom, Microsoft Research Cambridge, United Kingdom, and the University of Florence.

Marcello Pelillo is a Full Professor of Computer Science at Ca' Foscari University in Venice, Italy, where he leads the Computer Vision and Pattern Recognition group. He held visiting research positions at Yale University, McGill University, the University of Vienna, York University (UK), the University College London, and the National ICT Australia (NICTA). He has published more than 150 technical papers in refereed journals, handbooks, and conference proceedings in the areas of pattern recognition, computer vision and neural computation. He serves/has served on the Editorial Board for the journals *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Pattern Recognition*, *IET Computer Vision*, *Brain Informatics*, and is on the Advisory Board of the International Journal of Machine Learning and Cybernetics. He organized several conferences and workshops as Program Chair and is General Chair for ICCV 2017. He is/has been scientific coordinator of several research projects, including SIMBAD, an EU-FP7 project devoted to similarity-based pattern analysis and recognition whose activity is described in a new Springer book. He is a Fellow of the IEEE and a Fellow of the IAPR.

Horst Bischof received the MS and PhD degrees in computer science from the Vienna University of Technology in 1990 and 1993, respectively. In 1998, he got his Habilitation (venia docendi) for applied computer science. Currently, he is the vice rector for research at the Graz University of Technology and a professor at the Institute for Computer Graphics and Vision at the same university. His research interests include object recognition, visual learning, motion and tracking, visual surveillance and biometrics, medical computer vision, and adaptive methods for computer vision, where he has published more than 650 peer reviewed scientific papers. He has received several (19) awards among them are the 29th Pattern Recognition Award in 2002; the main price of the German Association for Pattern Recognition DAGM in 2007 and 2012, the Best Scientific Paper Award at the BMCV 2007, the BMVC Best Demo Award 2012, and the Best Scientific Paper Awards at the ICPR 2008, ICPR2010, PCV 2010, AAPR2010, and ACCV 2012. He is a member of the European Academy of Sciences.