Graz University of Technology

Institute for Computer Graphics and Vision

Institute of Paper, Pulp and Fiber Technology

Dissertation

Context-Aware Random Decision Forests for Object Detection and Semantic Segmentation

Peter Kontschieder

Graz, Austria, 2013

Thesis supervisors

Prof. Dr. Horst Bischof

Prof. Dr. Marcello Pelillo

If you understand what you're doing,
you're not learning anything.

Abraham Lincoln

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ……………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ……………………………………………..
date (signature)

Abstract

In this Thesis we introduce several ways to incorporate contextual information in a popular machine learning framework known as *random decision forests*. Traditionally, this type of learning algorithm processes training samples in a completely independent way and does not assume relations between them. While such an assumption greatly facilitates the overall learning problem, it might be suboptimal in cases where the data to be learned from exhibits certain structure as e.g. in images.

We focus on two core computer vision problems, object detection and semantic segmentation where the underlying data is assumed to be structured. With structured we refer to directed relations of spatially adjacent samples, which are typically mapped on a regular pixel/voxel grid and exhibit a large degree of correlation. From a more general point of view, this type of data structure is also reflected in the semantics and is commonly termed as contextual information. As machine learning plays a crucial role in modern computer vision systems, we stress the importance of exploiting this additional information about the data composition. Consequently, learning algorithms used for computer vision problems should take the structure of the data into account and learn correlations between samples not only from their representations but also from spatial and semantic arrangements.

The two main parts of this Thesis describe how random forests can be extended to integrate contextual information into the learning process in order to improve performance for object detection and semantic image segmentation tasks, respectively. For object detection, we investigate how the learner can be extended by using local shape information as a novel feature cue or how instances of an object category can be detected by inferring mutual compatibilities of individual data samples. Moreover, we introduce context-sensitive forests which exploit contextual cues by learning from intermediate predictions of the data. Such intermediate predictions can be considered as feedback loop into the training data and model early stages of the predictor outputs.

The second part deals with semantic image segmentation and we show how output predictions of random forests can be made interdependent in order to overcome the previously mentioned independence assumption of data samples. First, we present a way to directly encode this information in the output by introducing structured-class labels, which are local label patches respecting label semantics and their spatial distribution. In another approach, we show how predictor outputs can be interrelated and fed back into the training process by analyzing geodesic connectivity features, capturing semantic paths in a contrast-sensitive manner.

In both parts, we provide comprehensive experimental evaluations for the respective contributions, supporting our claims that context-awareness in the random forest machine learning algorithm leads to significant improvements of the results. For object detection, we evaluated on standard benchmark data bases like the TUD pedestrian data base, the ETHZ shape data base and the INRIA horses data base. For semantic segmentation, we evaluated data bases like the MSRCv2, CamVid driving videos, Labelled Faces in the Wild or Kinect data bases. Finally, we provide a summary about the Thesis and conclude with an outline for possible future work.

Keywords. Decision trees, ensemble methods, random forests, object detection, semantic segmentation.

Kurzfassung

Die vorliegende Doktorarbeit behandelt maschinelles Lernen mittels *Random Decision Forests* und führt mehrere neue Methoden vor, mit denen kontextuelle Informationen in diesem Lernverfahren berücksichtigt werden können. Obwohl diese meist stark korrelieren, behandelt dieses Lernverfahren Trainingsdaten gängiger Weise vollkommen unabhängig voneinander, womit die Informationen aus wechselseitigen Beziehungen ungenutzt bleiben. Eine unabhängige Betrachtung der Daten vereinfacht zwar im Allgemeinen das Lernproblem, führt jedoch zu einer ineffizienten Verwendung der Trainingsdaten, im Speziellen wenn diese eine charakteristische Struktur aufweisen, wie es etwa bei Bilddaten der Fall ist.

Unter der Annahme, dass die ihnen zugrunde liegenden Daten strukturiert sind, werden zwei zentrale Probleme der Bildverarbeitung neu bearbeitet: Objekterkennung und semantische Segmentierung. Strukturiert bedeutet dabei, dass räumlich benachbarte Datenpunkte, die auf ein reguläres Pixel-, bzw. Voxelraster abgebildet werden, ein hohes Maß an orientierungsabhängiger Korrelation aufweisen. Allgemeiner verstanden bildet diese Struktur auch die semantische Bedeutung in Bildern ab und wird als kontextuelle Information bezeichnet. Lernalgorithmen spielen in modernen Bildverarbeitungssystemen eine immer bedeutendere Rolle und sollten Korrelationen in den Daten sowohl auf Basis der Repräsentation als auch der kontextuellen Beziehungen erlernen.

Die beiden Hauptteile dieser Arbeit, Objekterkennung und semantische Segmentierung mit Random Decision Forests, beschreiben spezifische Erweiterungsmöglichkeiten zur Integration von kontextuellen Informationen für die Verbesserung der jeweiligen Ergebnisse. Zur Objekterkennung wird eine konturbasierte Datenrepräsentation eingeführt und gezeigt wie Objekte als Summe von gelernten, lokalen Fragmenten erkannt werden können. Weiters wird ein neues Verfahren zur Objekterkennung vorgestellt, welches Instanzen von gesuchten Objekten durch gemeinsame Analyse von paarweisen Kompatibilitäten detektiert, basierend auf Bestimmungen von Random Decision Forests. Ein weiteres entwickeltes Verfahren greift direkt in den Lernprozess ein, indem man dem Algorithmus während des Lernens bereits den Zugriff auf eigene, vorläufige Ergebnisse zu den Trainingsdaten ermöglicht, was wiederum den Zugriff auf kontextuelle Information für den restlichen Lernprozess erlaubt.

Im zweiten Teil dieser Forschungsarbeit werden Random Decision Forests erweitert, um verbesserte Resultate für semantische Segmentierung von Bilddaten zu erzielen. Als erster Ansatz wird eine Modifikation beschrieben, die strukturierte anstelle von skalarer Klassifikationsergebnisse bereitstellt. Somit können räumlich benachbarte Datenpunkte auch in deren Ergebnisraum miteinander verschränkt werden. In einer weiters vorgestellten Methode wird aufgezeigt, dass sich eine Rückführung von vorläufigen Ergebnissen in die Trainingsdaten positiv auf die Segmentierungsergebnisse auswirkt. Die Zwischenergebnisse werden einer kontrast-sensitiven Analyse und Glättung unterzogen, um eine Verschränkung im Ergebnisraum zu erzielen.

In beiden Hauptteilen werden umfangreiche, experimentelle Evaluierungen durchgeführt und die jeweiligen Ergebnisse mit bekannten Methoden aus der Literatur verglichen und diskutiert. Quantitative Analysen wurden zur Objekterkennung auf Datensätzen wie etwa TUD pedestrians, ETHZ shapes oder INRIA horses durchgeführt während semantische Segmentierung auf Datensätzen wie MSRCv2, CamVid driving videos, Labelled Faces in the Wild oder KINECT depth data evaluiert wurde. Generell kann gesagt werden, dass die Verwendung von kontextueller Information zu einer deutlichen Verbesserung der Resultate führt und das sowohl für Objekterkennung als auch für semantische Segmentierung. Auf Basis der durchgeführten Forschung ergeben sich zahlreiche Anknüpfungspunkte für zukünftige Untersuchungen, die am Schluss der Arbeit aufgezeigt werden.

Schlüsselwörter. Entscheidungsbäume, Ensemble-Methoden, Objekterkennung, semantische Segmentierung.

Acknowledgments

Taking the decision to attend a doctoral school may be almost done on an individual basis, but for better or worse does have an effect on others. This room is dedicated to express my gratitude to persons who inspired and supported me during my PhD studies.

I want to thank my supervisor Horst Bischof who supported me throughout my studies, taught me the necessary skills, induced inspirational thoughts and provided me space to work at his institute. My honest thanks go also to my second supervisor Marcello Pelillo, who, after first meeting him at ICPR'2010 triggered a re-orientation in the way I approached the field of visual computing. Thank you for hosting me twice as a visiting researcher at your institute at the Ca'Foscari University in Venice, and also for introducing me to one of your post doc researchers, Samuel Rota Bulò.

Samuel, this Thesis has emerged in the way it is because of your prolific mentoring and astonishing knowledge about our research field, yielding to both, a great friendship and wonderful publications I am proud of – Thank you! I am also very grateful for the countless number of fruitful discussions with many colleagues of the ICG, especially with Peter M. Roth, Thomas Mauthner, Samuel Schulter and my office mate Sabine Sternig. In a parallel researcher life I was dealing with all sorts of weird image data, mostly stemming from serially sectioned paper fiber specimen. I thank the guys from the Institute for Paper, Pulp and Fiber technology for providing me with challenging tasks to be performed on their data and also for ensuring my fundings over the past years. In particular, I am thanking Wolfgang Bauer and Ulrich Hirn for putting confidence in me to handle these tasks – May the fiber be with you! Moreover, in 2012 I had the great opportunity to visit Microsoft Research in Cambridge as an intern and could work and discuss with Antonio Criminisi, Pushmeet Kohli, Jamie Shotton and many more brilliant researchers, for which I am very thankful.

I am grateful for the unconditioned support of my friends who were accompanying me over many years, found time for every adventure and when necessary, reminded me about all the great things in life, going beyond the regular pixel grid. Special thanks go to Arno, Christian, Cornelia, Daniel D., Daniel K., Denise, Gerhild, Harald, Jürgen, Lissi, Lydia, Margret, Simon A., Simon S., Tomislav and Wolfgang! Finally, and with the words of Hermann Hesse – *If I know what love is, it is because of you* – I want to thank my partner and friend Nina for simply being amazing.

This Thesis was funded by the Institute for Paper, Pulp & Fiber Technology and the Institute for Computer Graphics and Vision at Graz University of Technology. The work was mainly supported by the Research Studios Austria Project μSTRUCSCOP (project number 818651) and the Austrian Science Fund (FWF) project Fibermorph (project number P22261-N22).

Contents

List of Figures

List of Tables

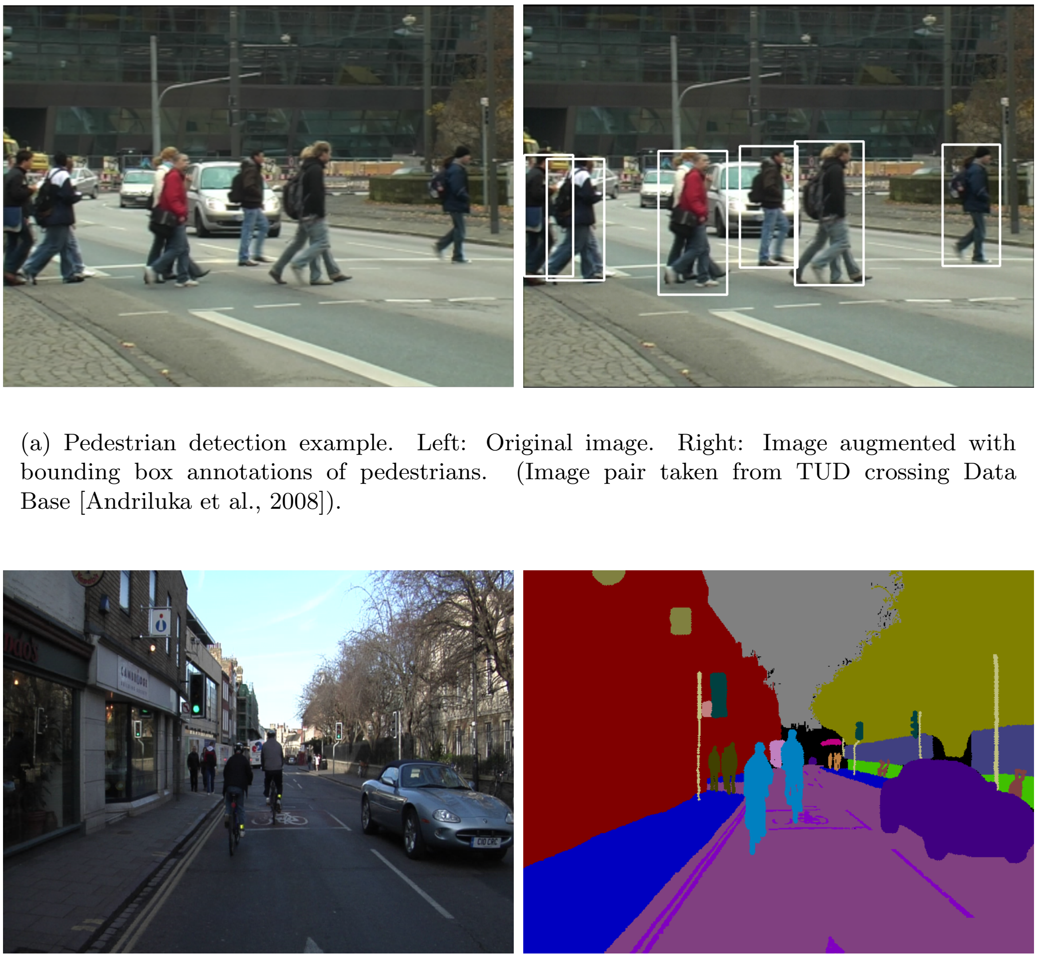
Chapter 1

Introduction

Contents

This Thesis investigates theoretical properties and provides practical insights for a supervised machine learning algorithm denoted as *random decision forest*, with emphasis on its customization for two classical computer vision problems, namely object detection and semantic image segmentation (see illustrations in Figure 1.1). With object detection, we refer to the task of category-specific object localization in a (previously unseen) test image. For instance, we want to find all pedestrians present in a given image by labelling them using an enclosing bounding box. Semantic image segmentation on the other hand has the goal to assign a categorical label to each pixel of an image under test. As an example, we might be given an image of a road scene where we want all pixels to be properly labelled as street, building, car, pedestrian, and so on. In such a way, semantically labelled images provide the basis for high-level reasoning about what is actually going on an image and is a key component to artificial scene understanding.

Consequently, one might ask about the relevance of machine learning when it comes to such computer vision problems. Reconsidering the previous examples we can immediately see that they both perform a *labelling* task on the test images. However, both described labelling tasks differ quite significantly in terms of their output domains. As for the bounding box labelling example (i.e. object detection) of pedestrians in an image, we want real-valued outputs that encode width and height dimensions as well as their positions in the image domain. In contrast, all we care about in semantic image labelling is to assign one out of possibly many categorical labels to each pixel in the image. Luckily, it turns out that such labelling tasks are exactly what may be accomplished by off-the-shelf machine learning algorithms such as support vector machines (SVM), boosting, or decision trees. However, before we argue for the choice of selecting exactly random decision trees as



(a) Pedestrian detection example. Left: Original image. Right: Image augmented with bounding box annotations of pedestrians. (Image pair taken from TUD crossing Data Base [Andriluka et al., 2008]).

(b) Semantic Image Labelling example. Left: Original image. Right: Corresponding, color-coded label image (Image pair taken from CamVid Data Base [Brostow et al., 2008]).

Figure 1.1: Illustrations for computer vision problems: Top row shows object detection task and bottom row shows semantic segmentation task.

learners, we conduct a trip to their historical evolution.

1.1 Historical Development of Learning with Decision Trees

”The ability to learn is a hallmark of intelligent behavior, so any attempt to understand intelligence as a phenomenon must include an understanding of learning.” With these words, Quinlan [Quinlan, 1986] gave reasons for the prominence of artificial intelligence as a research field, ever-increasing since its emergence in the mid 1950’s. In his seminal work, the focus was put on ”a family of machine learning systems that have been used to build knowledge-based systems of a simple kind". However, early ideas of top-down, inductive learning techniques like the Concept Learning System framework (CLS) of [Hunt et al., 1966] were attempting to minimize rather complex compositions of cost measures involving lookahead strategies (similar to minimax), requiring substantial amounts of computation. Independently, research in [Friedman, 1977] presents a recursive partitioning algorithm based on the Kolmogorov-Smirnov distance which is a measure for the separability of two distribution functions.

As one of their direct successors, the famous ID3 [Quinlan, 1979, Quinlan, 1983] provided a concept for inductive tree construction, built upon information theory driven evaluation functions and therefore replaced the computationally expensive idea of cost-driven lookaheads. Still, one of the remaining constraints was the restriction on the attributes (also referred to as feature space) which was resolved in the Analogue Concept Learning System (ACLS) of [Patterson and Niblett, 1983], giving permission to use numerical (integer-valued) and nonnumerical attributes. Reportedly, this extension allowed for the first time to apply decision trees to computer vision problems - for the task of classifying chocolate bars [Shepherd, 1983]:

The industrial vision problem described in [Shepherd, 1983] was posed as to correctly identify 12 different types of Black Magic chocolate bars, where each one was in a stable position but at arbitrary orientation. The feature space was designed from shape based statistics (e.g. compactness, circularity, elongation of corresponding silhouettes) extracted from boolean images with a resolution of 96 × 96 pixels. In the quantitative evaluation, the tree based classifiers were found to be competitive with respect to k-nearest neighbor (k-NN) or minimum distance classifiers (MDC), however with the advantages of low computational costs and the intelligibility of the solution.

Another successor of ID3, denoted as ASSISTANT, was introduced in [Kononenko et al., 1984]. It can be considered as generalization of ACLS as it extends to real-valued, nominal and also missing (or unknown) values. Further developments resulted in the well-known C4.5 [Quinlan, 1993] which is able to deal with multiple classes and different types of attributes (binary, nominal, (ordered) integer- and real-valued). As one of the most prominent works, Classification and Regression Trees (CART) [Breiman et al., 1984] provided a statistical view on the problem of decision tree training for both, classification and regression tasks, introducing the Gini-Index and Twoing for the test selection process.

Although fundamental research on decision trees was conducted and proven useful for some applications at that time, they were still restricted to quite low-dimensional data. However, with the rise of ensemble methods in the 1990's [Schapire, 1990, Freund, 1995] multiple decision trees were turned to decision forests. Averaging the outputs of multiple trees was introduced for the first time in [Amit and Geman, 1997] and showed significant improvements in terms of prediction accuracy (output variance reduction) but also better abilities to handle high-dimensional data.

In this line of research, the work in [Ho, 1998] investigated random partitioning of the input feature space and showed superior generalization properties. Another approach denoted as "bagging" (**b**ootstrap **agg**regat**ing**) was introduced in [Breiman, 1996a] and shows improvements when averaging results of decision trees that are trained on different random subsets of the original training data. Later, Breiman introduced the terminology of "Random Forests" [Breiman, 1999, Breiman, 2001] which have been popularly used for many machine learning tasks ever since.

1.2 Random Forests for Computer Vision Problems

Given the fact that there are many possible machine learning algorithms for handling computer vision problems - What makes decision trees our favourite choice of learning algorithm in this Thesis? In what follows, we discuss some of the prerequisites to our previously defined detection and segmentation goals while simultaneously remembering that there is no such thing as a universally optimal learning algorithm. Indeed, the *No free lunch theorem* [Wolpert, 1996] basically states that we should not get our hopes up for a single, consistently best learning algorithm. For example, it tells us that no matter how "good" or "bad" two different learning algorithms are, none of them will consistently outperform over the other when all target functions are equally likely. Ultimately, the no free lunch theorem encourages us to best possibly assess the circumstances of the target domain where the learning algorithm shall be applied. To this end, it is important to start with a definition of some basic conditions we agree to stick to throughout the rest of this Thesis.

We will only consider supervised learning problems, meaning that during the training process of a predictor function[1] we are given a set of training samples where each instance is a pair holding the actual object data (i.e. the input signal) and an associated, desired output signal (typically some label information obtained from ground truth data). This is in contrast to unsupervised or semi-supervised learning techniques which have no or only partially labelled data at their disposal. When it comes to decision forests, it is well known that their learning process efficiently scales to large-scale (and possibly high-dimensional) data sets while maintaining sustainable modelling capacities [Shotton et al., 2012]. This is of particular interest when dealing with image data, e.g. videos and images with high resolution are common and need to be processed in reasonable time.

Batch-based or *offline* learning is another constraint that we will restrict to, which assumes that all training data is available when we start with the learning process. This is opposed to *active* or *online learning* which allows additional, sequential arrival of training samples once the training process has already started. However, decision trees have been successfully introduced in the context of online learning [Saffari et al., 2009].

Random forests were termed as best off-the-shelf learning techniques in [Zhou, 2012] and are considered to be close to an ideal learner according to [Hastie et al., 2009]. In ad-

[1] With predictor functions we introduce a general terminology including the more specific classification or regression functions

dition, the work in [Caruana et al., 2008] empirically demonstrated that they outperform most state-of-the-art learners when it comes to handling high dimensional data problems. The seminal work of [Breiman, 2001] reported their efficiency for classification tasks. One of the most recent books on decision forests [Criminisi and Shotton, 2013] provides comprehensive information on how to exploit the core framework for tasks like regression, classification, semi-supervised learning, density estimation, manifold learning and active learning. In fact, they have been increasingly used for computer vision problems in the last couple of years [Marée et al., 2005, Lepetit and Fua, 2006, Criminisi et al., 2010, Bosch et al., 2007, Gall et al., 2011, Montillo et al., 2011, Shotton et al., 2008b, Glocker et al., 2012, Shotton et al., 2011, Shotton et al., 2012, Lim et al., 2013].

1.3 Motivation and Contributions of this Thesis

As shown earlier in this chapter, random decision forests are characterized by a series of elegant properties making them an important machine learning algorithm. However, when machine learning techniques are applied to computer vision problems, the available data typically exhibits certain properties which are often neglected throughout the training process. For instance, image data (2D-images, image sequences, volumetric image data) often exhibits high correlation between data points that are in turn arranged at a regular grid/voxel space. We can also consider it as structured data that naturally captures contextual information. In this sense, it seems desirable to include this contextual information into the learning process, i.e. one would like to exploit relations defined by terms of proximity, correlation and semantic context when learning from training data.

This Thesis is mainly concerned with how to integrate contextual information in the random forest framework to provide improved predictors for (mixed) classification and regression tasks in computer vision. Specifically, the Thesis is divided into three parts: First, we provide a general description of the random decision forest concepts and principles in Chapter 2, including a "core" theory that allows to formalize both, regression and classification problems. We provide specific instantiations of the general model to accomplish classification, regression and mixed classification and regression tasks.

The main contributions where we show the benefits from integrating contextual information in the learning process are divided into an object detection Part (I) and an object segmentation Part (II). The detection part is structured in three chapters:

- **Discriminative Learning of Contour Fragments for Object Detection.** This chapter introduces a way to efficiently learn shape fragments of fixed length together with their relative positions to object centroids of interest. To this end, we introduce a novel shape fragment descriptor which efficiently captures the local, contextual shape information and can furthermore be effectively learned together with its relative displacement to the object centroid within the random forest framework.

- **Evolutionary Hough Games for Coherent Object Detection.** Here, we provide a novel, game-theoretic approach for finding multiple instances of an object category, previously learned in a random forest. We exploit the information stored in the trees to setup a *payoff* matrix, that relates mutual compatibilities of multiple samples / patches of a test image to the previously learned object model. In such a way, we identify objects by means of the parts it is composed by and therefore jointly analyze their mutual, contextual relations.

- **Context-Sensitive Decision Forests for Object Detection.** In this chapter we introduce a new way to exploit contextual information provided by intermediate predictions of training/test samples, that are obtained from preceding nodes (or levels) of the trees. These preliminary predictions can be used as features in the subsequent training process, i.e. the learning process can be enhanced to learn about context and the semantic, spatial label distribution in the training data. A crucial problem is how to setup this contextual information which is tackled by introducing a prioritized way of training. Moreover, we introduce a general, loss-based formulation of the training process and provide a particular loss function which jointly optimizes for classification and regression problems.

The benefits of using contextual information in random forests for the task of image segmentation are described in the second part of this Thesis, which is structured in two chapters:

- **Structured Labels in Random Forests for Semantic Labelling and Object Detection.** In this chapter we investigate how to perform structured predictions in the output space of random forests. Conventional classification forests typically classify test samples with single (atomic) labels, resulting in noisy predictions when e.g. entire images shall be classified. However, semantic segmentation algorithms should ideally provide consistently and coherently labelled regions. For this task, we introduce a simple way to augment the output label space of random forests by analyzing and learning from structured class labels that implicitly encode the local arrangement of (multiple) labels given from ground truth information. The key benefits of our method are that we can learn semantically correct label transitions and compositions (i.e. we will only predict label configurations that exist in ground truth) and we can provide more robust segmentation results since also predictions in a local neighborhood can be derived from structured class labels. Moreover, we demonstrate that structured class labels also lead to significant improvements for the task of object detection when integrated in the concept of Hough forests.

- **Geodesic Forests for Learning Coupled Predictors.** Here, we demonstrate that an efficient, random forest based method is able to outperform a similar conditional random field (CRF) based method which involves the use of computational expensive inference algorithms. In our presented approach (and in contrast to CRF-based models), the posterior distribution completely factorizes over individual pixels, making it computationally very efficient. In order to obtain coherently segmented images for learned target categories, we present a novel way for coupling forest predictions back into the feature space of the classifiers where they learn from. The coupling mechanism interrelates pixel-pairs by analyzing long-range, semantic connectivity features derived from efficient, generalized geodesic distance transforms. In addition the chapter explores how MRF-like spatial coherence can be incorporated directly within the random forest training objective, yielding to superior segmentation quality as evinced in the experimental evaluation.

Chapter 2

Classification and Regression with Random Decision Trees

Contents

2.1 Notation

In this Thesis we denote *vectors* using boldface lowercase (e.g. $\mathbf{d}$, $\mathbf{u}$, $\mathbf{v}$) and *sets* by using uppercase calligraphic (e.g. $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{T}$) symbols. The sets of real, natural and integer numbers are denoted with $\mathbb{R}$, $\mathbb{N}$ and $\mathbb{Z}$, respectively. We denote by $2^{\mathcal{X}}$ the power set of $\mathcal{X}$ and by $\mathbb{1}[P]$ the indicator function returning 1 or 0 according to whether the proposition P is true or false. Moreover, with $\mathbb{P}(\mathcal{Y})$ we denote the set of probability distributions having $\mathcal{Y}$ as sample space. We denote by $\delta(x)$ the Dirac delta function (or Delta distribution). Additionally, $\mathbb{E}_{x \sim Q}\left[f(x)\right]$ denotes the expectation of $f(x)$ with respect to x sampled according to distribution Q.

Although we provide a general model of random forests that is not restricted to computer vision applications, the parameterisation for computer vision tasks discussed in this Thesis exploits the spatial arrangement of data, implicitly given by an image. To this end, we define a multi-channel image as a 3-dimensional matrix I where $I_{(\mathbf{u},c)}$ denotes the value at pixel position $\mathbf{u} = [u_1 \; u_2]^T)$ of channel c. Channels or *features* may e.g. include color image raw channel intensities, gradients, filter bank information, etc.. The set of images is denoted by $\mathcal{I}$.

2.2 Definition of Random Trees and Forests

In general, decision trees can be considered as tree-structured, predictive models for mapping (input) observations to certain target domains (e.g. categorical domains like person or car for classification tasks or continuous, real-valued domains for regression tasks). Decision trees are organized in a hierarchical way and may be interpreted as special instances of directed, acyclic graphs, consisting of nodes and edges. Here, we revisit a general description of the random forest framework in order to jointly capture classification and regression problems within a unified theoretic perspective. As we will demonstrate in the remainder of this Thesis, many existing random forest variants can be expressed in terms of the presented theory.

A binary *decision tree* is a tree-structured predictor[1] which, starting from its *root* (i.e. the origin of the tree), predicts about a sample that is routed until it reaches a leaf. At each internal node of the tree a decision is taken whether the sample should be forwarded to the left or right child, according to the outcome of a binary-valued function. In this Thesis we will only consider binary decision trees, i.e. each internal node will have exactly two child nodes and we will omit the *binary* attribute for brevity reasons.

In more formal terms, let $\mathcal{X}$ denote the input space, let $\mathcal{Y}$ denote the output space and let $\mathcal{T}$ be the set of decision trees. In its simplest form, a decision tree consists of a single node (a *leaf*) and is parameterised by a probability distribution $Q \in \mathbb{P}(\mathcal{Y})$ which represents the posterior probability of elements in $\mathcal{Y}$ given any data sample reaching the leaf, i.e. $P(y|x,t) = Q(y)$. We denote this tree as $\textsc{Lf}\left(Q\right) \in \mathcal{T}$. In this sense, an admittedly very rudimentary decision tree can consist of only a single leaf node, coinciding with the root node.

Otherwise, a decision tree consists of a node with a left and a right sub-tree. This node is parameterised by a *split function* $\phi : \mathcal{X} \to \{0,1\}$, which determines whether to route an arriving data sample $x \in \mathcal{X}$ to the left decision sub-tree $t_l \in \mathcal{T}$ (if $\phi(x) = 0$) or to the right one $t_r \in \mathcal{T}$ (if $\phi(x) = 1$). We denote such a tree as $\textsc{Nd}\left(\phi, t_l, t_r\right) \in \mathcal{T}$.

[1] We use the term predictor because we jointly consider classification and regression.

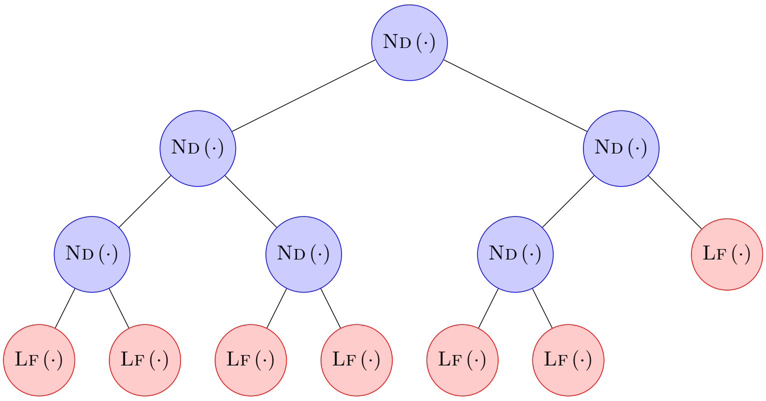


Figure 2.1: A single, binary decision tree $t \in \mathcal{T}$ with internal nodes $\textsc{Nd}\left(\cdot\right)$ in blue and leaf nodes $\textsc{Lf}\left(\cdot\right)$ in red.

Consequently, the set of decision trees can be compactly written as union of leaves and internal nodes

$$\mathcal{T} = \{\textsc{Lf}\left(Q\right) : Q \in \mathbb{P}(\mathcal{Y})\} \cup \left\{\textsc{Nd}\left(\phi, t_l, t_r\right) : \phi \in \mathcal{X} \to \{0,1\},\, t_{l/r} \in \mathcal{T}\right\} . \tag{2.1}$$

A *decision forest* is an ensemble $\mathcal{F} \subseteq \mathcal{T}$ of decision trees, which makes a prediction about a data sample by combining the single predictions gathered from the trees in the ensemble. The reasons for using multiple trees will be clarified later in this chapter, once the training and testing processes are described.

2.3 Learning/Training Process

Now that we know about the general structure of decision trees we describe the randomized training algorithm as presented in [Breiman, 2001]. The training procedure for a decision tree is devoted to determine the structure of the tree by proper selection of the split functions at the internal nodes and determination of the posterior probabilities to be stored in the leaves.

A random forest is created by independently training a set of random decision trees on random subsets of the training data $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$. The training procedure for finding the most compact (or smallest) decision trees for a given training set is known to be NP-complete [Hyafil and Rivest, 1976]. However, it has been empirically shown in [Murthy and Salzberg, 1995], that the expected classification costs for greedily induced trees as used in C4.5 and CART are consistently very close to those of optimal trees. The set of parameters to be optimized can e.g. determine the tree structure, the split functions at the internal nodes and the final predictions stored at the leaves. The goal of the training process is therefore to optimize the aforementioned parameters such that the prediction error on the training data is reduced and simultaneously achieve proper generalization such that non-seen test samples can be predicted with sufficient accuracy. In order to prevent overfitting problems, the search space of possible split functions is typically limited to a random set and a minimum number of training samples is required to grow a leaf node. Moreover, there exist a number of tree pruning strategies [Hastie et al., 2009], i.e. techniques applied after the growing phase that may alter the leaf parameterisation or truncate the trees by removing sub-trees. The main goal of these strategies is to decrease the generalization error by reducing the size of the trees while maintaining the prediction accuracy.

During the training procedure, each new node is fed with a set of training samples $\mathcal{Z} \subseteq \mathcal{D}$. If some stopping condition depending on $\mathcal{Z}$ holds, the node becomes a leaf and a density on $\mathcal{Y}$ is estimated based on $\mathcal{Z}$. Otherwise, an internal node is grown and a split function is selected from a pool of random ones in a way to minimize some pre-defined training error function on $\mathcal{Z}$. The selected split function induces a partition of $\mathcal{Z}$ into two disjoint sets, which in turn become the left and right childs of the current node where the training procedure is continued.

We will now write this training procedure in more formal terms. To this end we assume the existence of a function $\pi(\mathcal{Z}) \in \mathbb{P}(\mathcal{Y})$ providing a density on $\mathcal{Y}$ estimated from the training data $\mathcal{Z}$.

Loss function We define a *loss function* $L(\mathcal{Z} \,|\, Q) \in \mathbb{R}$ that penalizes wrong predictions on the independent and identically distributed (i.i.d.) training samples in $\mathcal{Z}$, when predictions are given according to a distribution $Q \in \mathbb{P}(\mathcal{Y})$ obtained from $\pi(\mathcal{Z})$. The loss function L can be further decomposed in terms of a loss function $\ell(\cdot|Q) : \mathcal{Y} \to \mathbb{R}$ acting on each sample of the training set in an (additive) way:

$$L(\mathcal{Z} \,|\, Q) = \sum_{(x,y)\in\mathcal{Z}} \ell(y \,|\, Q)\,. \tag{2.2}$$

We keep the definitions of both these functions general because they depend on the nature of the input and output spaces and on the specific application.

Split function Let $\Phi(\mathcal{Z})$ be a set of split functions randomly generated for a training set $\mathcal{Z}$ and given a split function $\phi \in \Phi(\mathcal{Z})$, we denote by $\mathcal{Z}_l^\phi$ and $\mathcal{Z}_r^\phi$ the sets obtained by splitting $\mathcal{Z}$ according to ϕ, i.e.

$$\mathcal{Z}_l^\phi = \{(x,y) \in \mathcal{Z} \,:\, \phi(x) = 0\} \tag{2.3}$$

and

$$\mathcal{Z}_r^\phi = \{(x,y) \in \mathcal{Z} \,:\, \phi(x) = 1\}\ . \tag{2.4}$$

In other words, given a split function $\phi \in \Phi(\mathcal{Z})$, we can obtain the parent training set with the union operator $\mathcal{Z} = \mathcal{Z}_l^\phi \cup \mathcal{Z}_r^\phi$ and $\mathcal{Z}_l^\phi \cap \mathcal{Z}_r^\phi = \emptyset$. The set of split functions $\Phi(\mathcal{Z})$ consists of models to separate the data in the input space $\mathcal{X}$ and is additionally parameterised with a set of parameters which however, in the general model description can be neglected but will be discussed later in this chapter.

Inductive training procedure Given the previously described components, we can now formalize the training procedure in terms of a recursive function $g : 2^{\mathcal{X}\times\mathcal{Y}} \to \mathcal{T}$, which generates a random decision tree from a given training set $\mathcal{Z}$ as argument:

$$g(\mathcal{Z}) = \begin{cases} \textsc{Lf}\,(\pi(\mathcal{Z})) & \text{if some stopping condition holds} \\ \textsc{Nd}\left(\phi, g(\mathcal{Z}_l^\phi), g(\mathcal{Z}_r^\phi)\right) & \text{otherwise}\,. \end{cases} \tag{2.5}$$

The optimal split function ϕ in the pool $\Phi(\mathcal{Z})$ is identified as the one minimizing the loss we incur given the node split

$$\phi \in \arg\min \left\{ L(\mathcal{Z}_l^{\phi'}) + L(\mathcal{Z}_r^{\phi'}) \,:\, \phi' \in \Phi(\mathcal{Z}) \right\} \tag{2.6}$$

where we compactly write $L(\mathcal{Z})$ instead of $L(\mathcal{Z}|\pi(\mathcal{Z}))$, i.e. the loss on $\mathcal{Z}$ obtained with predictions driven by the probability density estimate obtained from $\pi(\mathcal{Z})$.

Finally, the stopping condition that is used in (2.5) to determine whether to create a leaf or to continue branching the tree typically consists in checking if $|\mathcal{Z}|$, i.e. the number of training samples at the node, or the loss $L(\mathcal{Z})$ are below some given thresholds, or if a maximum depth of the tree is reached. The corresponding pseudo-code of the training procedure can be found in Algorithm TrainDT.

2.4 Inference/Testing Process

Given a decision tree $t \in \mathcal{T}$, the associated posterior probability of each element in $\mathcal{Y}$ given a sample $x \in \mathcal{X}$ is determined by finding the probability distribution Q, parameterising the leaf that is reached by x when routed along the tree. This is compactly presented with the following definition of $P(y|x,t)$, which is inductive in the structure of t:

$$P(y\,|\,x,t\,) = \begin{cases} P(y\,|\,x,t_l) & \text{if } t = \textsc{Nd}\,(\phi,t_l,t_r) \text{ and } \phi(x) = 0 \\ P(y\,|\,x,t_r) & \text{if } t = \textsc{Nd}\,(\phi,t_l,t_r) \text{ and } \phi(x) = 1 \\ Q(y) & \text{if } t = \textsc{Lf}\,(Q)\ . \end{cases} \tag{2.7}$$

As can be seen, Equation (2.7) corresponds to routing a sample x through a tree t, by taking directions to the left and right subtrees t_l, t_r according to the binary results of the respective, internal node split functions $\phi(x)$ until it reaches a leaf node. Once the leaf node is reached, the prediction takes place according to its stored probability distribution $P(y\,|\,x, t = \textsc{Lf}\,(Q)\,) = Q(y)$.

Finally, the combination of the posterior probabilities derived from

```
Procedure TrainDT($\mathcal{Z}, d$)
  Input: Training sample set $\mathcal{Z} \subseteq \mathcal{X} \times \mathcal{Y}$
  Input: Current tree depth $d$
  Input: Stopping condition: Max depth $d_{\max}$ or minimum number of samples $m$
         required for a leaf
  Output: An inductively grown decision tree $t$
  if $|\mathcal{Z}| < m$ or $d \geq d_{max}$ then
      return $t \leftarrow$ LF $(\pi(\mathcal{Z}))$
  else
      $s \leftarrow \infty$;
      $\Phi \sim \mathcal{U}$ ;   // randomly sample split functions $\Psi$ according to uniform
      distribution $\mathcal{U}$
      for $\phi' \in \Phi$ do
          $\{\mathcal{Z}_l^{\phi'}, \mathcal{Z}_r^{\phi'}\} \leftarrow \mathcal{Z}$ ;        // partition $\mathcal{Z}$ according to split $\phi'$
          $s' = L(\mathcal{Z}_l^{\phi'}) + L(\mathcal{Z}_r^{\phi'})$;
          if $s' < s$ then
              $s = s'$;
              $\phi = \phi'$ ;                   // store best split parameterising
      TrainDT($\mathcal{Z}_l^{\phi}, d+1$);
      TrainDT($\mathcal{Z}_r^{\phi}, d+1$);
      // trains left and right decision trees ND $\left(\phi, g(\mathcal{Z}_l^{\phi}), g(\mathcal{Z}_r^{\phi})\right)$
```

all the trees in a forest $\mathcal{F} \subseteq \mathcal{T}$ can be done by an averaging operation [Amit and Geman, 1997, Breiman, 2001], yielding a single posterior probability for the whole forest:

$$P(y|x,\mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{t \in \mathcal{F}} P(y|x,t)\,. \tag{2.8}$$

Alternatively, combining the output of the individual trees in the ensemble can be done in a multiplicative way

$$P(y|x,\mathcal{F}) = \frac{1}{Z} \prod_{t \in \mathcal{F}} P(y|x,t)\,, \tag{2.9}$$

where Z is a partition function that assures proper normalization. While simple averaging as in Equation (2.8) produces 'soft' decisions, the multiplicative combination enforces 'harder' ones. In both settings however, each input sample $x \in \mathcal{X}$ is assigned an element of the output space $\mathcal{Y}$ *independently* from all other samples.

2.5 Bagging, Multiple Trees and the Effect of Randomness

A general problem faced in machine learning is the so-called bias-variance trade off, typically arising from the inability of given predictor models to simultaneously minimize two specific types of prediction errors. To this end, prediction errors can be decomposed into additive error terms that are denoted as *bias* and *variance* errors, respectively (plus a data-dependent noise term). The bias term describes the difference between the desired, true output and the expected prediction of a model. The error conducted due to variance can be considered as the variation in the output of multiple predictors for a data sample.

The idea behind bagging is to average results from many high-variance but rather low-biased predictors, by training each predictor on individually sampled training data sets. To this end we remark that a decision tree can largely reduce bias, only by growing it to sufficient depth. Also, with bagging the trees are identically distributed (i.d.) so the expectation for the bias over the forest is the same as the expectation for the bias of individual trees. For random forests [Breiman, 2001], the idea of bagging has been extended by building ensembles of de-correlated decision trees, yielding to a reduction of the variance of prediction results. To illustrate the importance of the de-correlation between trees and its interplay with the reduction in error due to variance, we consider a tree in the forest $\mathcal{T}$ as identically but not necessarily independently distributed random variable $w_i \sim \mathcal{N}(\mu, \sigma^2)$, $w_i \in \mathcal{W}$ and $i = 1, \ldots, |\mathcal{T}|$. Here, μ is the mean and σ^2 is the variance, defining a Gaussian. As shown in [Hastie et al., 2009], we can now analyze the variance of the expected mean for correlated samples using

$$\mathbb{E}\left[\mathcal{W}\right] = \mathrm{Var}(\mathcal{W}) + (\mathbb{E}\left[\mathcal{W}\right])^2 \tag{2.10}$$

and the positive, pairwise correlation

$$\rho = \frac{\mathbb{E}\left[(w_i - \mu)(w_j - \mu)\right]}{\sigma^2} \text{ with } \rho > 0\,. \tag{2.11}$$

After some algebraic manipulation we obtain

$$\mathrm{Var}\left(\frac{1}{|\mathcal{T}|}\sum w_i\right) = \rho\sigma^2 + \frac{1-\rho}{|\mathcal{T}|}\sigma^2\,, \tag{2.12}$$

establishing a connection between the correlation ρ, the variance σ^2 and the number of trees in the forest $|\mathcal{T}|$. As the latter increases, i.e. the more trees are grown, the second term in (2.12) will vanish and therefore the limiting term is the correlation between trees for a given σ^2.

In addition to bagging, variance reduction of decision trees is obtained by employing randomization in the learning process. This causes de-correlation between the trees and is achieved by random selection of features used in the split nodes. In the following section, the selection mechanism is described in more detail for specific parameterizations of the random forest model.

2.6 Specific Parameterisations

Given the general definition of decision trees as well as the training and inference processes from the previous sections, we now focus on specific instantiations. In particular, we start by describing two of the most widely used instances, namely *classification* and *regression* trees. Moreover, we will present a recently developed type of random forest denoted as *Hough Forest* [Gall and Lempitsky, 2009], which is a joint classification and regression forest customized to the task of object detection. For all types, we will discuss the parametrization of the general models by providing a detailed description of their respective splitting and selection criterion functions (expressed in terms of loss functions above). A typical split function selection criterion commonly adopted for classification and regression is information gain or the Gini impurity. Since this interpretation is predominant in the literature [Criminisi et al., 2012], we will adopt to this setting. However, the equivalent counterpart in terms of loss can be obtained by using a log-loss, i.e. $\ell(y|Q) = -\log(Q(y))$ for the information gain or $\ell(y|Q) = 1 - Q(y)$ for the Gini impurity.

2.6.1 Classification Trees

The classification task is probably the most common application of random decision trees. Here, the goal is to associate each test sample to a discrete, categorical, unordered *class label*. Classification trees exhibit many desired properties like inherent multi-class capability, they generalize well to new test samples, tend not to overfit and are robust to label noise [Breiman, 2001]. Another interesting property is that they provide a probabilistic output as opposed to standard SVM. Many computer vision problems have been formulated as classification tasks, e.g. semantic image labelling seeks for a per-pixel, categorical classification of an image, or image categorization aims for recognizing the category of a scene captured in the image (see illustrations in Figure 1.1 and Figure 2.2).

According to the problem statement in [Criminisi et al., 2012], the classification task within the random decision tree model may be summarized as

> Given a labelled training set, learn a general mapping which associates previously unseen test data with their correct classes.

Consequently, after accomplishing the training process and applying the inference rules of (2.7) for a given test sample $\mathbf{x} \in \mathcal{X}$ and a single tree $t \in \mathcal{T}$ results in an (empirical) posterior probability distribution $P(y|\mathbf{x}, t)$ from where the discrete class label

$$y^* = \arg\max_{y \in \mathcal{Y}} P(y|\mathbf{x}, t) \tag{2.13}$$

is selected as the most represented element in the sample space of $\mathcal{Y} = \{1, \ldots, k\}$.

Figure 2.2: Image categorization as classification task. Two examples for category 'Mountain scene'.

In a straightforward manner, given an ensemble of trees $\mathcal{F} \subseteq \mathcal{T}$ and their combination $P(y|\mathbf{x}, \mathcal{F})$ using (2.8) or (2.9), the final class label is selected as

$$y^* = \arg \max_{y \in \mathcal{Y}} P(y|\mathbf{x}, \mathcal{F}) \,. \tag{2.14}$$

Split functions The split functions are responsible for separating training data during the tree growing phase and deciding where to route a sample during the testing phase. In such a way, each internal node has to store its own parameters $\theta = \{\psi, \tau, \vartheta\}$ for the split function $\phi(\cdot|\theta) \in \Phi(\mathcal{Z})$. Here, ψ defines a geometric data separation model (a hyperplane, general surface, etc.) to partition the data in the input domain. The second parameter $\tau \in \mathbb{R}$ defines a threshold quantity for the inequalities that are typically used as separation functions. Finally, $\vartheta(\mathbf{v})$ can be termed as feature selection function (a feature sub-space selection as introduced in [Ho, 1998]) that (randomly) selects a subset of dimensions of the input data vector $\mathbf{v}$ such that

$$\vartheta : \mathbb{R}^d \to \mathbb{R}^{d'} \quad \text{with} \quad d' << d \,. \tag{2.15}$$

As an example, let the input space $\mathcal{X}$ be a collection of data samples $\{\mathbf{v}_i\}_{i=1}^N, \forall i : \mathbf{v}_i \in \mathbb{R}^d$. Then, a split function with a linear separation model can be simply defined by

$$\phi(\mathbf{v}, \theta) = \mathbb{1}\left[\vartheta(\mathbf{v})^T \cdot \psi > \tau\right] , \tag{2.16}$$

where $\psi \in \mathbb{R}^{d'}$ is a vector of length d', that parametrizes a general, oriented hyperplane. Moreover, τ can be interpreted as margin to this hyperplane, allowing to study max-margin properties of the random forest model, as recently conducted in [Leistner et al., 2009, Criminisi et al., 2012].

As mentioned earlier in this chapter, computer vision applications may benefit from the spatial arrangement of data inherently available to images, i.e. individual pixels of an image are anchored to a grid structure. To this end, we assume the input domain $\mathcal{X}$ to be a set of image patches. Each patch $\mathbf{x}$ is defined by a pair $\mathbf{x} = (\mathbf{u}, I) \in \mathcal{X}$, i.e. the patch center position $\mathbf{u}$ in a multi-channel image $I \in \mathcal{I}$. Moreover, the separation model is often reduced to very simple types like axis aligned hyperplanes, also known as decision stumps. As a consequence, ψ vanishes and the feature selection function $\vartheta(\mathbf{x})$ reduces to selecting a single feature channel c in the multi-channel images. Therefore, we can introduce a more compact parameter set $\theta'_i = \{\Delta\mathbf{v}_i, c_i\}, i = 1, 2$ where $\Delta\mathbf{v}_i$ is a displacement vector relative to the patch center, resulting in the following split functions as used in [Shotton et al., 2008b, Kontschieder et al., 2011]:

$$\begin{aligned}
\phi^{(1)}(\mathbf{x}|\theta', \tau) &= \mathbb{1}\left[I_{(\mathbf{u},0)+(\Delta\mathbf{v}_1,c_1)} > \tau\right], &(2.17)\\
\phi^{(2)}(\mathbf{x}|\theta'_1, \theta'_2, \tau) &= \mathbb{1}\left[I_{(\mathbf{u},0)+(\Delta\mathbf{v}_1,c_1)} - I_{(\mathbf{u},0)+(\Delta\mathbf{v}_2,c_2)} > \tau\right], &(2.18)\\
\phi^{(3)}(\mathbf{x}|\theta'_1, \theta'_2, \tau) &= \mathbb{1}\left[I_{(\mathbf{u},0)+(\Delta\mathbf{v}_1,c_1)} + I_{(\mathbf{u},0)+(\Delta\mathbf{v}_2,c_2)} > \tau\right], &(2.19)\\
\phi^{(4)}(\mathbf{x}|\theta'_1, \theta'_2, \tau) &= \mathbb{1}\left[\left|I_{(\mathbf{u},0)+(\Delta\mathbf{v}_1,c_1)} - I_{(\mathbf{u},0)+(\Delta\mathbf{v}_2,c_2)}\right| > \tau\right]. &(2.20)
\end{aligned}$$

Training objective functions From the general definition of the training procedure in Section 2.3 we can see, that the goal of the training process is to find the optimal split function parametrization such that the loss computed on the resulting child training sets is minimized (see Equation (2.6)). Here, we provide a reformulation in terms of an objective function

$$\phi = \underset{\phi' \in \Psi(\mathcal{Z})}{\arg\max}\ G(\mathcal{Z}|\phi') \tag{2.21}$$

which seeks to maximize the *information gain* G

$$G(\mathcal{Z}|\phi) = H(\mathcal{Z}) - H(\mathcal{Z}|\phi), \tag{2.22}$$

using the Shannon entropy defined over probability mass functions for discrete random variables W with alphabet $\mathcal{W}$, where $P(w) = Pr(W = w)$, $w \in \mathcal{W}$

$$H(W) = -\sum_{w \in \mathcal{W}} P(w) \log\left(P(w)\right). \tag{2.23}$$

Information gain is a commonly agreed objective used for classification tree learning [Quinlan, 1983, Quinlan, 1993, Criminisi et al., 2012]. The reason why it suits well is that each branch in a tree, i.e. each newly added internal node seeks to reduce the uncertainty about the class label prediction, measured in an information-theoretic sense. In such a way, the prediction confidence increases as the hierarchical structure of the tree approaches the leaves where the final per-tree prediction takes place.

Easy to see, for our task the alphabet is defined by the output space $\mathcal{Y}$ and the probability mass function is obtained from $\pi(\mathcal{Z})$ estimated over the training data labels of $\mathcal{Z}$. Due to reasons of simplicity and speed, normalized histograms over the class labels contained in the training set are typically used as estimator function.

Then, the information gain about the label distributions $\left\{\mathcal{Z}_l^{\phi}, \mathcal{Z}_r^{\phi}\right\}$ obtained from a split $\phi \in \Phi(\mathcal{Z})$ is defined as

$$G(\mathcal{Z}|\phi) = H(\mathcal{Z}) - \sum_{i\in\{l,r\}} \frac{|\mathcal{Z}_i^{\phi}|}{|\mathcal{Z}|} H(\mathcal{Z}_i^{\phi})\,. \tag{2.24}$$

The first part of Equation (2.24) does not depend on the split function which is why we can simplify (2.21) to

$$\begin{aligned} \phi &= \underset{\phi'\in\Psi(\mathcal{Z})}{\arg\max} \left\{ H(\mathcal{Z}) - \sum_{i\in\{l,r\}} \frac{|\mathcal{Z}_i^{\phi'}|}{|\mathcal{Z}|} H(\mathcal{Z}_i^{\phi'}) \right\} & (2.25)\\ &= \underset{\phi'\in\Psi(\mathcal{Z})}{\arg\min} \left\{ \sum_{i\in\{l,r\}} \frac{|\mathcal{Z}_i^{\phi'}|}{|\mathcal{Z}|} H(\mathcal{Z}_i^{\phi'}) \right\}\,. & (2.26) \end{aligned}$$

A closer look at Equation (2.26) reveals that it all boils down to a sum of weighted entropies computed on the posterior distributions induced by the split function parametrization. In fact, lower entropy means that the uncertainty about the outcome of an experiment for a certain event decreases, i.e. its associated probability increases. In case of classification tree learning, we observe the effect of narrowing down the choices for particular class labels while passing the hierarchy of the tree. The weights $\frac{|\mathcal{Z}_i^{\phi'}|}{|\mathcal{Z}|}$, $i \in \{l, r\}$ combined with the respective entropies control the balance for the induced splits. For example, a split that trivially removes a single element from $\mathcal{Z}$ such that $|\mathcal{Z}_l^{\phi'}| = 1$ and $|\mathcal{Z}_r^{\phi'}| = |\mathcal{Z}| - 1$ would result in $H(\mathcal{Z}_l^{\phi'}) = 0$ for the left subtree but high weights for $H(\mathcal{Z}_r^{\phi'})$. Additionally, and as already mentioned in context of the stopping conditions for the growing process, a minimum cardinality for the resulting child sets can be enforced.

The recent work in [Nowozin, 2012] investigates the quality of information gain estimation under the light of finite training sample sets. In particular, the problems of conventionally used plug-in estimators are addressed and properties like bias and systematic underestimation of true entropy is reported. To this end, [Nowozin, 2012] introduces several alternatives of which the Grassberger entropy estimators are investigated in more detail for classification and regression tasks.

Another way to formulate the split function criterion exploits the so-called *Gini-impurity*, which was originally introduced in [Breiman et al., 1984]. The objective of this criterion is to measure how often an element in the data is misclassified, given the current probability density estimation $Q = \pi(\mathcal{Z})$ over the labels reaching a node. Then, the Gini impurity is defined as

$$\hat{H}(\mathcal{Z}) = \sum_{i=1}^{|\mathcal{Y}|} q_i(1 - q_i) = \ldots = 1 - \sum_{i=1}^{|\mathcal{Y}|} q_i^2 \tag{2.27}$$

and can be simply plugged into (2.26) instead of the entropy term.

The optimization task in the training process is typically conducted in a greedy way by evaluating all split functions in $\Phi(\mathcal{Z})$ and keeping the one that induces the lowest loss. This is mainly due to the following reasons. For example, the popularly used information gain is not differentiable with respect to the binary split function parameters ϕ so standard optimization cannot be directly applied (alternatively, a differentiable sigmoidal function was used in [Montillo et al., 2013] to represent the boundary partitioning function). Another problem relates to overfitting and therefore reduce the desired generalization properties of the trees. Moreover, exhaustive search over the entire parameter space may become prohibitively expensive, which is why the training process is driven by a random component that generates ad-hoc parameterisations $\phi' \in \Phi(\mathcal{Z})$ by uniformly sampling over all possible parameters. Therefore, it is not necessary to pre-compute all possible parameterisations, making the optimization process very fast and efficient.

2.6.2 Regression Trees

With different parametrization of the general model, random decision trees can be used to predict continuous, real-valued variables. Termed as *regression trees*, they are typically used as non-linear regressors with the goal to discriminatively learn the relations between independent, continuous, multi-variate inputs and dependent, continuous and multivariate outputs. In the sequel we will demonstrate that regression trees not only share advantageous properties with classification trees in terms of efficiency but also that some of the parametrization concepts can be adapted in a straightforward manner. To this end, we start with a motivation by showing some exemplary computer vision applications like medical image analysis [Criminisi et al., 2010] and head-pose estimation [Fanelli et al., 2011a] in Figure 2.3.

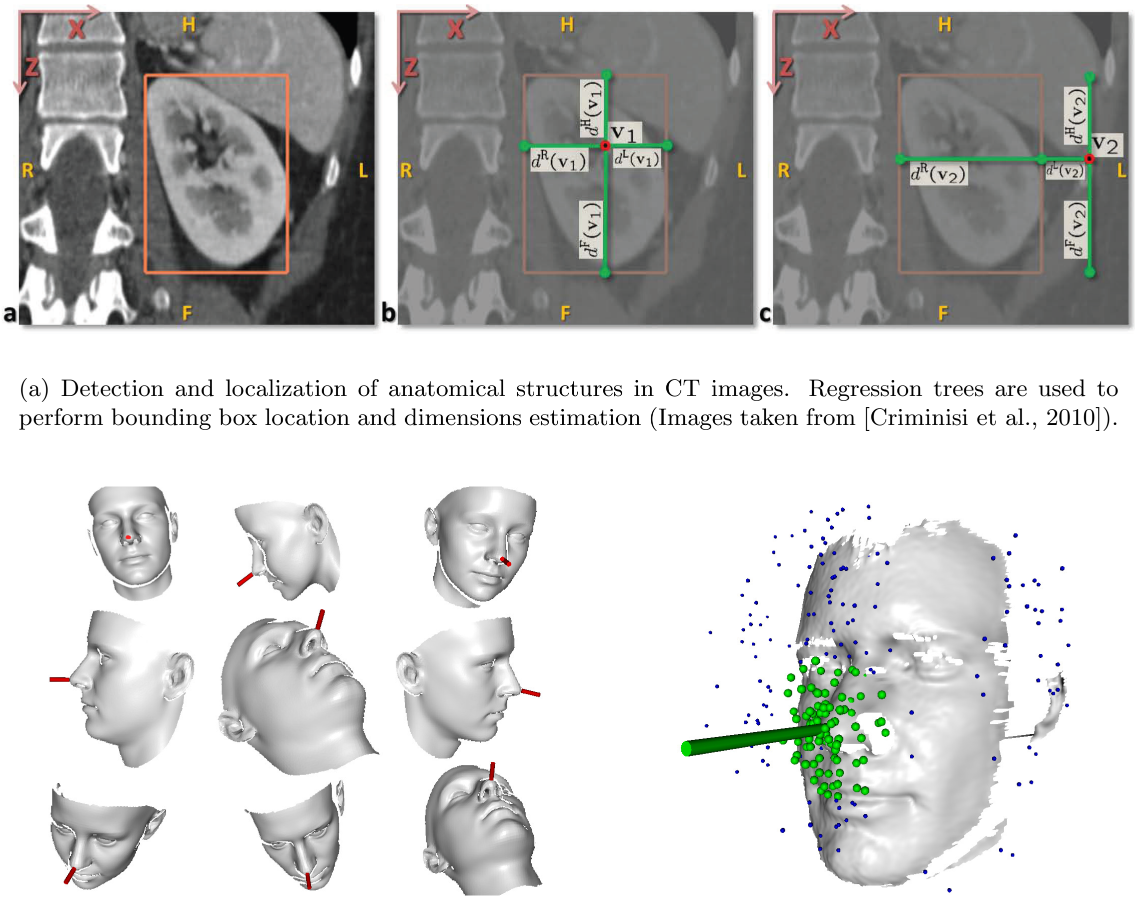
Again, we start by providing an informal problem definition for the regression task within the random decision tree model from [Criminisi et al., 2012]:

> Given a labelled training set, learn a general mapping which associates previously unseen independent test data with their correct continuous prediction.

For the moment, let us assume we are given a set of regression trees $\mathcal{T}$, learned from random subsets of the training data $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$. Then, for an unseen test sample $\mathbf{x} \in \mathcal{X}$ we want to obtain a prediction in form of a general probability density estimation over the (continuous) output variable $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^n$ such that

$$P(\mathbf{y}|\mathbf{x}, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} P(\mathbf{y}|\mathbf{x}, t)\,. \tag{2.28}$$

Obviously, if the sought prediction is a continuous-valued, multi-variate label $\mathbf{y} \in \mathcal{Y}$, also the training data must be in a form $\mathcal{Z} \subseteq \mathbb{R}^d \times \mathbb{R}^n$, i.e. $\forall\,(\mathbf{x}, \mathbf{y}) \in \mathcal{Z} : \mathbf{x} \in \mathbb{R}^d, \mathbf{y} \in \mathbb{R}^n$.



(a) Detection and localization of anatomical structures in CT images. Regression trees are used to perform bounding box location and dimensions estimation (Images taken from [Criminisi et al., 2010]).

(b) Real-time head-pose estimation from ($2\frac{1}{2}$D) range images. Regression trees are used to learn a mapping from training data (left) to predict head pose illustrated by green cylinder (right). (Images taken from [Fanelli et al., 2011a]).

Figure 2.3: Illustrations for computer vision applications possibly using regression trees.

Split functions Similar to the classification tree case, the split functions serve the task to separate the training data during training and define the routing through the trees at test time. In such a way, predictions obtained from regression trees may be seen as results of a sequence of hierarchically executed, (non-)linear regressors. A very simple way to perform the per-node regression for one-dimensional input and output values (e.g. to predict house prices as a function of living space) would be to use a generic polynomial of order k

$$f(x) = \sum_{i=0}^{k} w_i\, x^i \quad \text{with} \quad \mathbf{w} = [w_0\ \dots\ w_k]^T \tag{2.29}$$

Using this model, an internal tree node has to store the parameters $\theta = \{\mathbf{w}, \tau\}$, i.e. the parameters $\mathbf{w}$ for the polynomial and a threshold vector τ used in a split function

$$\phi(x, \tau) = \mathbb{1}\left[f(x) > \tau\right]. \tag{2.30}$$

Similar to the classification tree case where we provided a split function parametrization tailored to using images with pixels anchored to a grid structure, the works of [Criminisi et al., 2012, Fanelli et al., 2011a] used patch-based test functions computed over regions (or volumes) F_1, F_2 centered at positions $\mathbf{u}_1, \mathbf{u}_2$, respectively. Consequently, the split function can be defined as

$$\phi(\mathbf{x}_1, \mathbf{x}_2 | \theta'_1, \theta'_2, \tau) = \mathbb{1}\left[\left(\frac{1}{|F_1|}\sum_{\mathbf{v}_1 \in F_1} I_{(\mathbf{u}_1,0)+(\mathbf{v}_1,c_1)} - \frac{1}{|F_2|}\sum_{\mathbf{v}_2 \in F_2} I_{(\mathbf{u}_2,0)+(\mathbf{v}_2,c_2)}\right) > \tau\right], \tag{2.31}$$

such that the parameters $\theta'_i = \{F_i, c_i\}$ to be stored for the split function contain the locations/dimensions of the rectangular regions to be evaluated and the feature channel c where the evaluation shall take place. As can be seen, Equation (2.31) computes the difference between the average intensities in the regions of selected feature channels. Using regions instead of pixel-based differences makes the decision process more robust against noise. The computation of the average intensities can be done very efficiently using integral images [Viola and Jones, 2002]. Additionally, the feature channels for both regions are often select as $c_1 = c_2$, i.e. evaluation takes place on the same feature.

Posterior estimation Since we are interested in a regression model with probabilistic ouput, i.e. we want to know the confidence about the prediction, we need to provide a way for estimating the conditional probability $P(\mathbf{y}|\mathbf{x})$. For example, the work in [Criminisi et al., 2012, Appendix A] shows how parameters and the conditional density can be estimated when using a simple line regression model.

Here we focus on the multi-variate case, assuming that the output data follows a multi-variate Gaussian distribution, i.e.

$$\mathbf{y} \in \mathcal{Y} \propto \mathcal{N}(\mu, \Sigma) \tag{2.32}$$

that is parametrized with mean μ and covariance Σ. Indeed, this assumption is not too restrictive since the tree hierarchy allows to capture multi-modality as piece-wise Gaussians [Criminisi et al., 2010]. Moreover, using a parametric, Gaussian representation the information gain can be derived in a closed, analytic form.

Training objective functions Again, the goal of the training process is to find a parametrization $\phi' \in \Psi$ for each non-leaf node that minimizes the loss. Alternatively, we will again provide a formulation that maximizes the information gain G seeking for

$$\phi = \underset{\phi' \in \Psi(\mathcal{Z})}{\arg\max}\ G(\mathcal{Z}|\phi') \tag{2.33}$$

using

$$G(\mathcal{Z}|\phi) = h(\mathcal{Z}) - h(\mathcal{Z}|\phi)\,. \tag{2.34}$$

Now, we can employ the *differential entropy* $h(W)$ over the continuous random variable W with probability density $f(w)$ according to the definition in [Cover and Thomas, 2006]

$$h(W) = h(f) = -\int_{\Omega} f(w)\log f(w)dw\,, \tag{2.35}$$

where Ω is the support set of the random variable. Plugging (2.34) into Equation (2.33), we get

$$\begin{aligned} \phi &= \underset{\phi' \in \Psi(\mathcal{Z})}{\arg\max} \left\{ h(\mathcal{Z}) - \sum_{i\in\{l,r\}} \frac{|\mathcal{Z}_i^{\phi'}|}{|\mathcal{Z}|} h(\mathcal{Z}_i^{\phi'}) \right\} & (2.36)\\ &= \underset{\phi' \in \Psi(\mathcal{Z})}{\arg\min} \left\{ \sum_{i\in\{l,r\}} \frac{|\mathcal{Z}_i^{\phi'}|}{|\mathcal{Z}|} h(\mathcal{Z}_i^{\phi'}) \right\}\,. & (2.37) \end{aligned}$$

Moreover, since we assume $W_1, W_2, \ldots, W_n$ to be a multi-variate normal distribution, we can write

$$h(W_1, W_2, \ldots, W_n) = h(\mathcal{N}_n(\mu, \Sigma))\,. \tag{2.38}$$

With the definition of the probability density function for the normal distribution

$$f(\mathbf{w}) = \frac{1}{\left(\sqrt{2\pi}\right)^n |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{w}-\mu)^T\Sigma^{-1}(\mathbf{w}-\mu)} \tag{2.39}$$

where $|\Sigma|$ is the determinant of Σ, it can be shown that the differential entropy reduces to

$$\begin{aligned} h(\mathcal{N}_n(\mu, \Sigma)) &= \frac{1}{2}\log\left((2\pi e)^n |\Sigma|\right) & (2.40)\\ &= \underbrace{\frac{1}{2}\log\left(2\pi e\right)^n}_{\text{constant expression}} + \frac{1}{2}\log\left(|\Sigma|\right) & (2.41)\\ &\approx \log\left(|\Sigma|\right)\,, & (2.42) \end{aligned}$$

i.e. the differential entropy is determined by the determinant of the covariance matrix (up to a constant factor).

When we combine (2.42) with (2.37) we get

$$\phi = \underset{\phi' \in \Psi(\mathcal{Z})}{\arg\min} \left\{ \sum_{i \in \{l,r\}} \frac{|\mathcal{Z}_i^{\phi'}|}{|\mathcal{Z}|} \log(|\Sigma(\mathcal{Z}_i^{\phi'})|) \right\} . \tag{2.43}$$

and it follows that the best parametrization for a node split tries to minimize the sum of weighted, logarithmic determinants, induced by the partition. Finally, the leaf nodes store the parameters used in the regression task.

2.6.3 Hough Trees

Hough trees [Gall and Lempitsky, 2009, Gall et al., 2011] are a recently presented variant of random decision trees that can be used to perform a generalized Hough transform [Ballard, 1981] *together* with per-sample classification. In this sense, they combine some properties of classification and regression trees and are able to perform mixed, discrete- and continuous-valued predictions. In the sequel we will introduce the concept of Hough trees (and Hough forests) and explain how they can be used for class-specific object detection. Please note that there exist several extensions for action recognition [Yao et al., 2010], human pose estimation [Girshick et al., 2011] or multi-class object detection [Razavi et al., 2011], demonstrating their potential for the computer vision domain. The basic concept how Hough trees are used for object detection is illustrated in Figure 2.4.

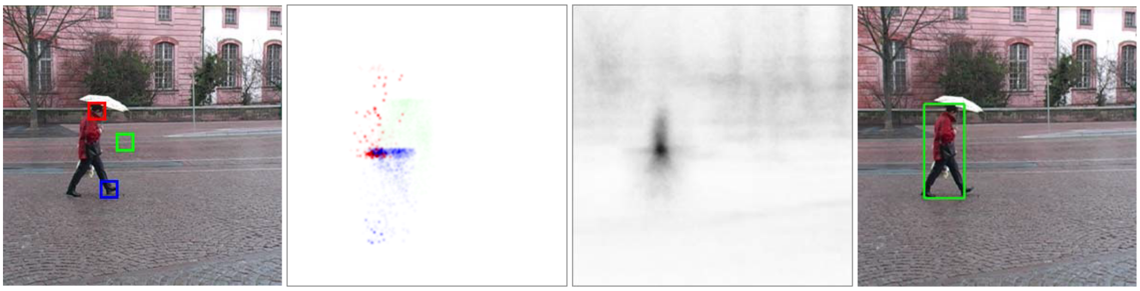


Figure 2.4: Illustration of Hough Forest concept for object localization. Leftmost image: Input image with example test patch locations (color-coded in red, green and blue). Second image: Voting (regression) information for object centroid corresponding to color-coded boxes from first image, cast into accumulative Hough voting space. Third image: Hough image containing accumulated votes from all evaluated patches. Rightmost image: Detection hypothesis as a result of maxima analysis performed in Hough image. (Images taken from [Gall and Lempitsky, 2009]).

In the line of the previous parameterisations discussed for decision trees, we give an informal definition as follows:

Given a labelled set of training images, learn a general mapping which associates previously unseen test data with both, correct categorical and continuous predictions.

Before we explain how Hough trees can be used for object detection, we briefly introduce the idea of the generalized Hough transform [Ballard, 1981] for object detection, which is a generalization of the original work in [Hough, 1959]. The basic idea behind the Hough concept is to abstract the input image into voting elements and transfer 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. center voting information). During inference, the predictors not only perform classification on test samples but also cast probabilistic votes in a generalized Hough-voting space that is subsequently used to obtain object center hypotheses.

More formally, the input space $\mathcal{X}$ for Hough trees is a set of patches and each of them is represented as a pair $(\mathbf{u}, I) \in \mathbb{Z}^2 \times \mathcal{I}$ where $\mathbf{u}$ is the center pixel position of the patch in image I. The output space $\mathcal{Y}$ is a set of pairs $(c_l, \mathbf{d})$ where $c_l \in \{0, 1\}$ is a binary class label 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 $(c_l, \mathbf{d}) \in \mathcal{Y}$, and can describe either a background sample ($c_l = 0$) or a foreground sample ($c_l = 1$). While the displacement vector is ignored for the background case, it indicates the object centroid corresponding to a foreground sample at location $\mathbf{u} + \mathbf{d}$ in image I. Please note that the output space $\mathcal{Y}$ encodes both, classification and regression information. Next, we will derive how the tree predicts the output for a given test sample $\mathbf{x} \in \mathcal{X}$ at a leaf node.

At a leaf node, the prediction is obtained from a density estimation function $\pi(\mathcal{Z})$, which generates the posterior distributions stored in the tree leaves. The provided distributions factorize in two marginal distributions, for the class labels and the displacement vectors, respectively. The marginal over the class labels is a discrete distribution, providing the probability of drawing a sample of a given class from $\mathcal{Z}$. To this end, let $q \in \mathbb{P}(\{0, 1\})$ be the marginal distribution over the class labels defined as

$$q^{\text{Class}}(c_l) = \frac{|\mathcal{Z}_{c_l}|}{|\mathcal{Z}|} \tag{2.44}$$

where $\mathcal{Z}_0$ and $\mathcal{Z}_1$ are the set of background and foreground samples in $\mathcal{Z}$ reaching a leaf, respectively.

At the same time, we are interested in the probability density marginal $q^{\text{Vote}}(\mathbf{v})$, estimated for location $\mathbf{v}$ over the voting information of the foreground samples using a Parzen window with a Gaussian kernel function

$$q^{\text{Vote}}(\mathbf{v}) = \frac{1}{|\mathcal{Z}_1|} \sum_{(c_l,\mathbf{d})\in\mathcal{Z}_1} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{||\mathbf{v}-\mathbf{d}||^2}{2\sigma^2}} , \tag{2.45}$$

where σ is the standard deviation of the Gaussian.

Consequently, a Hough tree stores a probability distribution $Q(y) = q^{\text{Class}}(c_l) \cdot q^{\text{Vote}}(\mathbf{v})$ in each leaf which is then used for prediction as defined in Equation (2.7). For example, given a new test sample $\mathbf{x} \in \mathcal{X}$, i.e. a patch centered at position $\mathbf{u}$ in image I, a Hough tree $t \in \mathcal{T}$ performs a probability estimate for a possible object center hypothesis $E(\mathbf{l})$ located at position $\mathbf{l}$ in the associated Hough image by evaluating

$$P(E(\mathbf{l})|\mathbf{x}, t) = q^{\text{Class}}(c_l = 1) \cdot q^{\text{Vote}}(\mathbf{l} - \mathbf{u}) . \tag{2.46}$$

As already known from the pure classification and regression trees, the Hough forest ensemble $\mathcal{T}$ estimates

$$P(E(\mathbf{l})|\mathbf{x}, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{t\in\mathcal{T}} P(E(\mathbf{l})|\mathbf{x}, t) . \tag{2.47}$$

Split function The split function originally used in [Gall and Lempitsky, 2009] is similar to those described for the classification trees and separates/routes samples based on the binary outcome of pixel-pair tests according to

$$\phi(\mathbf{x}|\Delta\mathbf{v}_1, \Delta\mathbf{v}_2, c, \tau) = \mathbb{1}\left[I_{(\mathbf{u},0)+(\Delta\mathbf{v}_1,c)} - I_{(\mathbf{u},0)+(\Delta\mathbf{v}_2,c)} > \tau\right] , \tag{2.48}$$

where $\Delta\mathbf{v}_i$ are offset vectors relative to the patch center $\mathbf{u}$ and τ is a threshold parameter. Differently from the split functions described for classifcation trees, the selected feature channel c does not change for the respective pixel positions.

Training objective functions Obviously, the training procedure of Hough trees also aims for reducing the prediction error on the training set and generalizing to unseen test data. The key difference to the training procedures of standard decision trees as used for classification or regression is that the training objective minimizes either the *class-label uncertainty* or the *offset uncertainty* based on a randomized decision. In what follows we will describe both criteria, starting with the class-label uncertainty.

The class-label uncertainty, which we denote with $U^{\text{Class}}(\cdot)$, is essentially the weighted entropy measure over the class label distributions as already described in Equation (2.26). Since Hough trees are originally designed to handle only binary classification problems, it can be unrolled to

$$U^{\text{Class}}(\mathcal{Z}) = |Z_0|H(Z_0) + |Z_1|H(Z_1) . \tag{2.49}$$

For the regression tree case, we have provided a parametric way to estimate the probability density of continuous valued variables and moreover, showed how the training objective can be formulated in an information-theoretic sense. In Hough trees, the training objective when minimizing the offset uncertainty $U^{\text{Vote}}(\cdot)$ is not explicitly formulated in an information theoretic sense but is instead optimized by minimizing the variance in the set of centroid voting vectors reaching a node (which is the optimal estimator for a zero-mean Gaussian). This can be formalized as

$$U^{\text{Vote}}(\mathcal{Z}) = \frac{1}{|Z_1|} \sum_{(c_l,\mathbf{d})\in\mathcal{Z}_1} \left|\left|\mathbf{d} - \overline{\mathbf{d}}\right|\right|^2 \tag{2.50}$$

where

$$\overline{\mathbf{d}} = \frac{1}{|Z_1|} \sum_{(c_l,\mathbf{d})\in\mathcal{Z}_1} \mathbf{d} \tag{2.51}$$

is the mean voting vector of the foreground samples. Please note that $U^{\text{Vote}}(\cdot)$ considers only foreground training samples and all background samples are ignored.

As a result, the training objective function seeks for the optimal parameters

$$\phi = \underset{\phi'\in\Psi(\mathcal{Z})}{\arg\min} \left\{ \sum_{i\in\{l,r\}} U^{\star}(\mathcal{Z}_i^{\phi'}) \right\} . \tag{2.52}$$

where $\star = \{\text{Class}, \text{Vote}\}$ randomly selects (according to a uniform distribution) either the class-label uncertainty of (2.49) or the offset uncertainty of (2.50).

Another way to formulate the objective function was presented in [Okada, 2009]. Instead of selecting between the respective uncertainties a combined score is suggested as follows

$$\phi = \underset{\phi'\in\Psi(\mathcal{Z})}{\arg\min} \left\{ \sum_{i\in\{l,r\}} \left[U^{\text{Class}}(\mathcal{Z}_i^{\phi'})\right] + \alpha \max\left(q_{\mathcal{Z}}^{\text{Class}}(c_l = 1) - \gamma,\, 0\right) \sum_{i\in\{l,r\}} \left[U^{\text{Vote}}(\mathcal{Z}_i^{\phi'})\right] \right\} , \tag{2.53}$$

where α is a weight parameter and γ is an activation threshold for the regression score. As can be seen, the objective function in (2.53) will only include the regression part when the foreground probability exceeds γ. Moreover, the regression term additionally becomes more important according to the weight parameter α as the foreground probability increases.

While the objective function presented in (2.53) incorporates the offset uncertainty based on the foreground probabilities induced by a split function parametrization, the work in [Fanelli et al., 2011b] proposes to include the regression term as a function of the tree depth d with

$$\phi = \underset{\phi' \in \Psi(\mathcal{Z})}{\arg\min} \left\{ \sum_{i \in \{l,r\}} \left[U^{\text{Class}}(\mathcal{Z}_i^{\phi'}) \right] + \left(1.0 - e^{-\frac{d}{\lambda}} \right) \sum_{i \in \{l,r\}} \left[U^{\text{Vote}}(\mathcal{Z}_i^{\phi'}) \right] \right\} . \tag{2.54}$$

Here, λ is a parameter that controls the steepness of the exponential function weighting the contribution of the voting uncertainty.

2.7 Extensions / Different Variants of Decision Trees

2.7.1 Extremely Randomized Trees

The extremely randomized trees (Extra-Trees) algorithm was introduced by [Geurts et al., 2006] and focuses on the reduction of optimization load in the parameter selection process for the split functions and minimizes the correlation between pairs of trees (see Section 2.5). Instead of performing a (greedy) optimization over the possible parameters of the split function pool $\Psi(\mathcal{Z})$, Extra-Trees provide an extremely randomized setting where most parameters for the split function are selected in a completely randomized way. In the extreme case where all parameters are selected randomly, this has the effect that no node optimization has to be performed. However, typically the loss on the training data is much higher and the prediction confidence is lower [Criminisi et al., 2012].

2.7.2 Entangled Decision Trees

Entangled decision trees were recently introduced in [Montillo et al., 2011] and are a special instance of classification trees that are allowed to use intermediate node classification results within the split functions (in addition to standard image features). The basic idea is to exploit the steadily increasing prediction confidence induced by the tree hierarchy already *within* the training process and therefore learn the semantic relations of class labels in their geometric context. In such a way, the resulting entanglement random forest augments the possible feature channels with a classification feature, that is consequently refined by the class posterior distributions according to the progress of the trained subtree. In their experiments on semantic segmentation of medical images the authors report significant improvements of the segmentation results.

2.7.3 Random Ferns

So far we have introduced random trees which impose a sequence of hierarchically arranged tests on a sample, where the reachability of each node depends on the outcome of the respective parent node. Random ferns are in contrast to this hierarchy and instead perform a sequence of tests on each sample, independent on the result of the split function in the previous node. In such a way, a random fern could be considered as a random decision tree using the same parameters for all split functions at the same tree level.

In the original work of [Özuysal et al., 2007], the authors introduce ferns as a fast alternative to decision trees and motivate them from a naive Bayesian point of view. In particular, each fern is regarded as a semi-naive Bayesian combination of a series of binary tests (cf. split functions as used for classification trees). This has the effect to bypass a complete independence assumption between the simple binary tests and can instead model available correlation in the training data using a compact representation. Random ferns were initially used for classification (keypoint recognition was treated as classification task) but there exist extensions for regression [Dollár et al., 2010] and also online-learning [Villamizar et al., 2012, Godec et al., 2010].

Here, we consider random ferns as special instances of decision trees with the aforementioned restriction that, for a certain depth level d of the tree, each split function $\phi_d(\cdot)$ performs the same test. Typically, split functions operate only on single feature channels of the training data (as described for the classification trees in Section 2.6.1). This has the effect that axis-aligned hyperplanes are employed in the split function, requiring random ferns to be grown deeper in comparison to standard classification trees which may partition the input space differently according to the selected path in the tree.

2.8 Summary

In this chapter we provided an introduction to supervised learning using random decision trees. More specifically, we have given a general definition of binary, random decision trees and forests by describing their composition, structure as well as a recursive way to formalize training and testing processes, respectively. An important task when approaching computer vision problems is to find suitable parameterisations in order to accomplish classification, regression or mixed classification/regression problems. To this end, we have introduced a general formalism that is able to account for all of the aforementioned parameterisations and provided specific instantiations for state-of-the-art algorithms like e.g. Hough Trees.

Part I

Object Detection with Random Decision Trees

Chapter 3

Discriminative Learning of Contour Fragments for Object Detection

Contents

In this chapter we discuss contour-based object detection using random forests. The focus is put on shape fragment based data representation and how we can effectively learn it in a random forest model. In the learning phase, we interrelate local shape descriptions (fragments) of the object contour with the corresponding spatial location of the object centroid, using the Hough forest as introduced in the previous chapter in Section 2.6.3. We introduce a novel shape fragment descriptor that abstracts spatially connected edge points into a matrix consisting of angular relations between the points. Our proposed descriptor fulfills important properties like distinctiveness, robustness and insensitivity to clutter. During detection, we hypothesize object locations in a generalized Hough voting scheme. The back-projected votes from the fragments allow to approximately delineate the object contour. We evaluate our method on well-known data bases like the ETHZ shape data base or the INRIA horses, emphasizing on the importance of properly designed data representation for the random forest learning framework.

3.1 Introduction

Object localization in cluttered images is a big challenge in computer vision. Typical methods in this field learn an object category model, e.g. from a set of labeled training images and use this model to localize previously unseen category instances in novel images. The two dominating approaches in this field are either sliding window [Felzenszwalb et al., 2010] or generalized Hough-voting [Ballard, 1981] based. The final detection results are mostly returned as bounding boxes highlighting the instance locations, but some methods also return accurate object outlines.

In general, the detection approaches can additionally be divided into appearance and contour based methods. Appearance-based approaches first detect interest points and then extract strong image patch descriptors from the local neighborhoods of the detected points using versatile features like color, texture or gradient information. In contrast, contour-based methods exhibit interesting properties like low sensitivity to illumination changes or variations in color or texture. It is also well-established in visual perception theory [Biederman and Ju, 1988] that most humans are able to identify specific objects even from a limited number of contour fragments without considering any appearance information.

3.1.1 Related Work

3.1.1.1 Appearance-based models

The main paradigm in this field is the bag-of-visual-words model introduced in [Sivic and Zisserman, 2003] where the authors represent the object class as a collection of orderless local image descriptors [Mikolajczyk and Schmid, 2005]. In [Lazebnik et al., 2006], this approach was extended by incorporating spatial arrangements of features in a pyramidal matching setup and showed improved detection performance. Another related, appearance based representative is the Implicit Shape Model (ISM) [Leibe et al., 2004]. An ISM is a class-specific codebook of local appearance descriptors where each descriptor additionally contains information about the relative object centroid location. During recognition, extracted image patches are matched to the codebook and cast probabilistic votes in the sense of the generalized Hough transform [Ballard, 1981] to hypothesize object locations. However, unsupervised integration of a large number of object parts in a generative codebook approach involves a large-scale clustering problem and a time-consuming matching step. To avoid these problems, some researches replaced the generative codebooks by discriminative learning of the object model parts. A very prominent representative for discriminative learning are Hough Forests [Gall and Lempitsky, 2009] as described in the previous chapter. Another approach incorporating the Hough transform in a discriminative learning approach was introduced in [Maji and Malik, 2009]. They focus on identifying object parts showing both, high repeatability and consistency in their locations and then assign higher weights in the voting step. An extension of this work for the hypothesis verification phase was presented in [Ommer and Malik, 2009] by extending conventional, spatial Hough voting with a scale dimension. They analyze the voting space by clustering of voting lines to reduce the initial candidate detection set.

3.1.1.2 Contour-based models

In [Fergus et al., 2003], the authors showed a learning approach which incorporates shape (besides appearance information) as a joint spatial layout distribution in a Bayesian setting for a limited number of shape parts. [Ferrari et al., 2006] partition image edges of the object model into groups of adjacent contour segments. For matching, they find paths through the segments which resemble the outline of the modeled categories. In a later approach [Ferrari et al., 2008] they investigate how to define contour segments in groups of k approximately straight adjacent segments (kAS) as well as how to learn a codebook of pairs of adjacent contours (PAS) from cropped training images [Ferrari et al., 2007] in combination with Hough-based center voting and non-rigid thin-plate spline matching.

[Ravishankar et al., 2008] use short line fragments, favoring curved segments over straight lines allowing certain articulations by splitting edges at high curvature points. In [Leordeanu et al., 2007], a recognition system that models an object category using pairwise geometrical interactions of simple gradient features was introduced. The resulting category shape model is represented by a fully connected graph with its edges being an abstraction of the pairwise relationships. The detection task is formulated as a quadratic assignment problem. [Shotton et al., 2005] and [Opelt et al., 2006] simultaneously introduced similar recognition frameworks based on boosting contour-based features and clutter sensitive chamfer matching. Both methods construct a codebook and employ shape features in a star-constellation around the object centroid. The contour fragments used in the codebook are selected by an AdaBoost learning stage after clustering similar fragments. Each fragment is aware of its position along the object contour and holds information about its spatial displacement to the object centroid. For recognition, they compare contours learned from the training set to the edges found in the test images using clutter sensitive Chamfer matching and cast probabilistic votes in a generalized Hough space manner to find object hypotheses. The main differences are that [Shotton et al., 2005] use a single fragment per weak detector and a grid structure to localize the object center while [Opelt et al., 2006] have a variable number of fragments and mean shift is applied to find the centroid localization where.

An approach using a cascade of boosted classifiers for object detection was proposed in [Smith et al., 2009]. They introduced a set of shape descriptors called Ray features, designed to capture irregular shapes as e.g. occurring in biological cell image data sets. A method for fragment grouping in a particle filter formulation was presented in [Lu et al., 2009] and obtained consistent models for detection. In [Bai et al., 2009], intra-class shape variations were captured within a certain bandwidth by a closed contour object model called shape band. In [Zhu et al., 2008b], the authors formulated object detection as a many-to-many fragment matching problem. They utilize a contour grouping method to obtain long, salient matching candidates which are then compared using standard shape context descriptors. The large number of possible matchings is handled by encoding the shape descriptor algebraically in a linear form, where optimization is done by linear programming. In a follow-up work [Srinivasan et al., 2010], promising results are reported for models, automatically obtained from training data instead of using a single category shape prototype. Again, their method relies on the availability of long, salient contours and has high complexity with detection times in the range of minutes per image. In [Riemenschneider et al., 2010], the authors perform object detection by partially matching detected edges to a prototype contour in the test images. In such a way, piecewise contour approximations and error-prone matches between coarse shape descriptions at local interest points are avoided. Another approach proposed in [Yarlagadda et al., 2010] aims on grouping mutually dependent object parts of uniformly sampled probabilistic edges. In their Hough voting stage, object detection is formulated as optimization problem which groups dependent parts, correspondences between parts and object models and votes from groups to object hypotheses. Another recent approach of [Payet and Todorovic, 2010] formulated object detection as mining repetitive spatial configurations of contours in unlabeled images. Their contours are represented as sequences of weighted beam angle histograms and are transferred into a graph of matching contours, with the maximum a posteriori multicoloring assignment being taken to represent the shapes of discovered objects. Moreover, we refer to the approach in [Amit and Geman, 1997] where randomized trees are used to perform shape recognition for handwritten digits. Opposed to our approach, the authors distinguish the shapes by recursively partitioning the shape space by growing binary classification trees using geometric arrangements among local topographic codes as splitting rules.

Lately, reasonable effort went into fusing appearance and contour based methods like [Toshev et al., 2010, Li et al., 2010]. In these works, the authors incorporate appearance information by either using superpixel segmentations together with statistics derived from their boundaries [Toshev et al., 2010] or exploit multiple figure-ground segmentations for the recognition task [Li et al., 2010].

3.1.2 Contributions

As can be seen in the comprehensive summary of related work in Section 3.1.1, many approaches were proposed that exploit shape information for object detection. However,

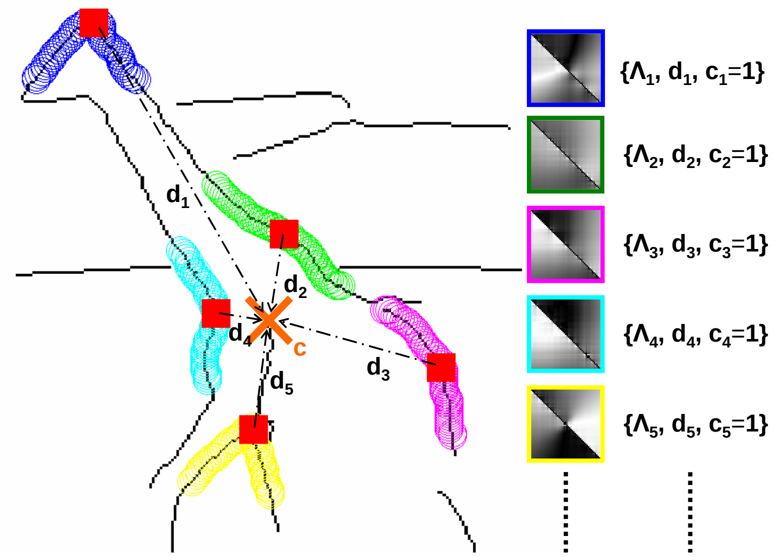


Figure 3.1: Illustration of our object category localization method. Local edge fragments are discriminatively trained in a Hough Forest analyzing a triple per fragment $\{\Lambda_i, \mathbf{d}_i, \mathbf{c}_i\}$, where Λ_i is our novel fragment descriptor matrix, $\mathbf{d}_i$ its corresponding center voting vector and $\mathbf{c}_i$ its class label. During testing descriptor matches vote for object centroid $\mathbf{c}$ to hypothesize object locations in the image. Best viewed in color.

we observed that many researchers first put an enormous effort into learning a suitable object model from training data [Shotton et al., 2005, Opelt et al., 2006, Ferrari et al., 2007, Ravishankar et al., 2008] but then neglect to directly apply this gained knowledge in the matching or verification phase. Instead, techniques like error-prone Chamfer matching are used to decide whether an object is present or not. We are convinced that a detection system benefits from an approach which inherently unifies the strengths of the individual methods instead of either solely relying on a particular technique or serially concatenating them. As a consequence, we propose to jointly learn a novel shape fragment description with its spatial location information about the object contour. For recognition, we can directly apply the learned knowledge in a generalized Hough voting manner.

A key issue of such an approach is to use a powerful local shape descriptor, which should fulfill various requirements like distinctiveness, robustness to clutter and noise, invariance and efficiency. Since common local shape descriptors are limited concerning these requirements, we propose a novel contour fragment descriptor which describes relative spatial arrangements of sampled points by means of angular information. As it is shown in the experiments in Section 3.3 our novel descriptor outperforms related methods like shape context in this scenario.

3.2 Discriminative Learning of Fragments

In this section we present our novel approach for object detection. As underlying representation we use connected and linked edges as it is described in Section 3.2.1. From the obtained edges we extract local fragment descriptions using a novel, discriminative descriptor that captures local angular information along edges (see Section 3.2.2). To learn a model from a set of labeled training images we train a Hough Forest on the obtained descriptors storing the relative location and scale of the fragments with respect to the object centroid for the positive training samples, as it is explained in Section 3.2.3. At run-time, we cast probabilistic votes for possible center locations of the target objects in a generalized Hough voting manner. The resulting local maxima in the Hough space serve as detection hypotheses. In Figure 3.1 we illustrate our proposed method.

3.2.1 Edge Detection and Linking

As a first step we extract edges of the input image. We use the Berkeley edge detector [Martin et al., 2004] in all experiments and link the edges into oriented, connected point coordinate lists. We want to explicitly stress the importance of point linking which has great impact on the obtained fragments, since different splits of T-junctions or gap closing yields very different fragments. The obtained lists of connected points state the basis for all subsequent steps which is why we introduce the following terminology, used throughout the rest of this chapter: We name all lists as *edges*, while we denote all extracted edge parts as *fragments*. In our method all analyzed fragments have the same length, consisting of exactly N points.

3.2.2 Shape Fragment Descriptor

To be able to discriminatively learn the shape of local, equal sized fragments, we need a powerful shape descriptor. Typical local fragment descriptors are shape context [Belongie et al., 2002], turning angles [Chen et al., 2008], beam angles [Payet and Todorovic, 2010], partial contours [Riemenschneider et al., 2010] or contour flexibility [Xu et al., 2009]. We propose a novel descriptor which, as shown in the experiments in Section 3.3, outperforms related methods. Our shape descriptor is related to the descriptor of Riemenschneider et al. [Riemenschneider et al., 2010] which also uses angles between fragment points. The major differences however are as follows: First, we employ another sampling strategy to define the angles and second, we differ in the selection of descriptor values. We analyze angles defined by lines connecting a reference point and the fragments' points. This is in contrast to [Riemenschneider et al., 2010] where only relative angles between points on the fragments are considered and in addition no fixed length of the fragment is assumed. Moreover, we are using a fragment-dependent reference point which is defined by the fragments' bounding box and therefore also contributes to a discriminative and local description of each considered fragment (see

Fig. 3.4). Our sampling strategy was carefully designed for usage within a discriminative learning framework and is crucial for reasonable performance since otherwise we would not be able to distinguish between different locations on regularly shaped object parts as e.g. the semi-circles in Figure 3.3.

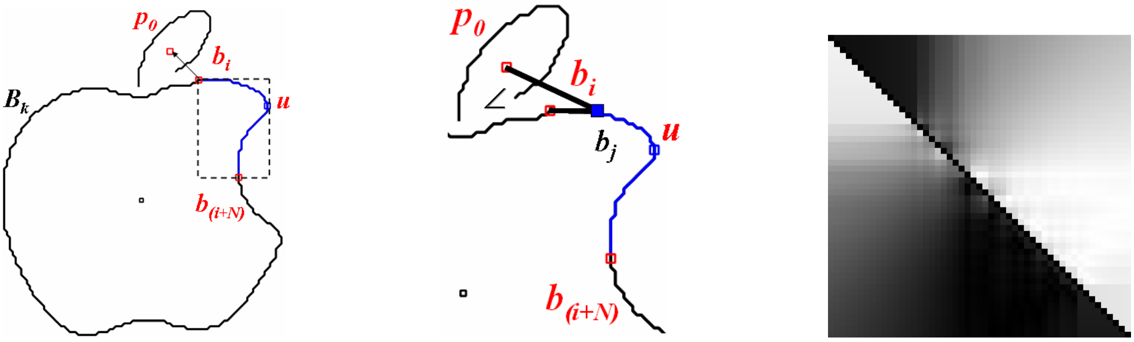


Figure 3.2: Illustration of fragment descriptor computation. Left: Apple-logo edge image (from ETHZ data set) with fragment of interest in blue with bounding box. Center: Zoomed edge fragment with construction sample for $\sphericalangle(\overline{\mathbf{b}_i\mathbf{b}_j}, \overline{\mathbf{b}_j\mathbf{p}_0})$. Right: Resulting fragment descriptor (cf. Figure 3.4).

We first outline the main definitions for extracting the fragment descriptors and then discuss the general properties of the obtained representation in detail. Let an individual *edge* B_i be defined as a sequence of linked points $B_i = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{M_i}\}$, where $M_i = |B_i|$ is the total number of connected edge points and $M_i \geq N$. We always analyze fragments of the same length N. Therefore, we compute $(M_i - N + 1)$ fragment descriptors for every individual edge B_i. For every *fragment* we define an $N \times N$ descriptor matrix Λ

$$\Lambda = \begin{pmatrix} 0 & \alpha_{12} & \cdots & \alpha_{1N} \\ \alpha_{21} & 0 & \cdots & \vdots \\ \vdots & & \ddots & \vdots \\ \alpha_{N1} & \cdots & \cdots & 0 \end{pmatrix} \tag{3.1}$$

with $diag(\Lambda) = \mathbf{0}$. Every entry $\alpha_{ij} (i \neq j)$ is defined by the angle between a line connecting the points $\mathbf{b}_i$ and $\mathbf{b}_j$ and a line from $\mathbf{b}_j$ to a reference point $\mathbf{p}_0$, which is defined by the upper left corner of the fragments' surrounding bounding box as illustrated in Figure 3.2. In such a way, we define

$$\alpha_{ij} = \sphericalangle(\overline{\mathbf{b}_i\mathbf{b}_j}, \overline{\mathbf{b}_j\mathbf{p}_0}) \quad \forall i,j = 1, \ldots, N \tag{3.2}$$

where $\mathbf{b}_i, \mathbf{b}_j$ are the i^{th} and j^{th} points on the respective fragment. Hence, we are mapping the fragment description onto the interval $[0, \pi]$. For a single fragment, the angles are calculated over all possible point combinations, yielding the descriptor matrix. Our proposed descriptor has a number of important properties which we discuss next.

Distinctiveness Most importantly, descriptors calculated from locations on the target object contour need to be discriminative to those generated from background data or clutter. Furthermore, we want the descriptors to be distinguishable from each other when they are calculated from the same object contour, but at different locations. This property is of particular interest when features are trained together with their spatial support. On the other hand, features that are computed at similar locations but from different training samples should result in similar representations. In other words, the descriptor should be able to capture intra-class variations and tolerate small perturbations in the training set.

In Figure 3.3, we show some fragments of geometric primitives like squares and circles and their corresponding descriptor matrices. Please note how the resulting patterns differ from each other. In Figure 3.4 we illustrate how our descriptor copes with intra-class variations of similar locations on the contours. The given samples are scaled to the same height while fixing the aspect ratios. All descriptors are computed on the fragments highlighted by the blue circles. Despite significant changes in the object contours, the resulting descriptors show similar patterns.

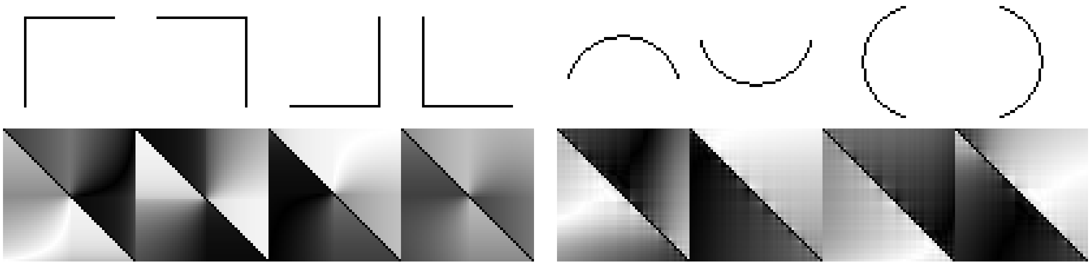


Figure 3.3: Selected contour fragment primitives (first row) and the corresponding, unique angular abstractions into the proposed fragment descriptor matrices (second row).

Efficacy Another important property of our descriptor is the efficacy of data abstraction. Using a measure to describe relations between connected points has multiple advantages over considering each pixel independently. It assists to identify discriminative fragments during training and simultaneously reduces noise since information is encoded in a redundant way.

Invariance Features are often classified according to their level of invariance to certain geometrical transformations. For example, features invariant to Euclidean transforms keep unchanged after applying translation and rotation. Increasing the degree of invariance generally decreases the distinctiveness and thus weakens the distinctiveness property. Our proposed descriptor is invariant to translation and encodes orientational information. In such a way it is not rotation invariant, which is however compensated in the discriminative learning training process.

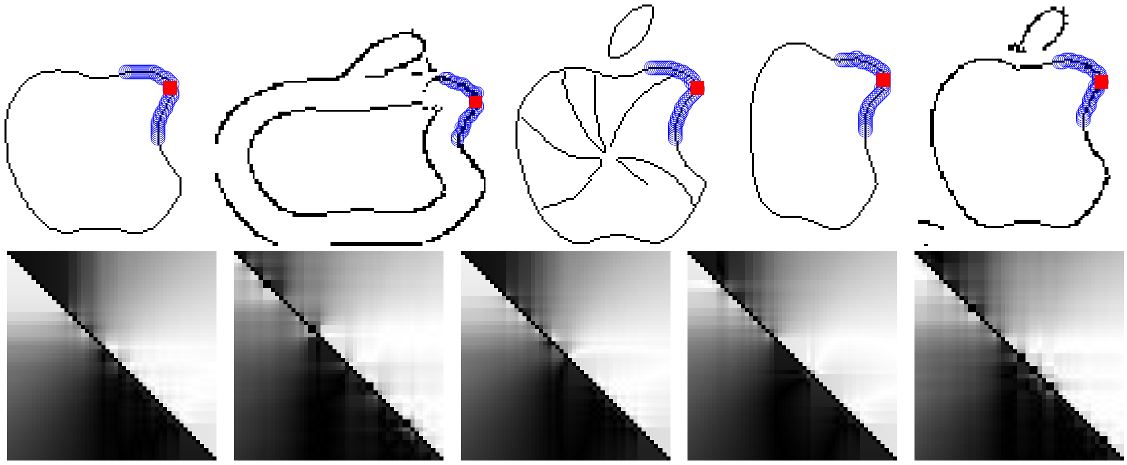


Figure 3.4: Mappings of intra-class variations for similar locations on object contours to descriptor matrices. Blue circles highlight the selected fragments and red squares indicate their center position.

Efficiency Features should be computable in an efficient manner. In our case, we can precompute the values for all possible pixel combinations, hence reducing actual feature composition to a lookup-operation which can be done in constant time for any fragment. Since the angular description in Equation (3.2) corresponds to a surjective mapping $f : Z \subset \mathbb{Z}^2 \to [0, \pi]$, we can provide a lookup-table (LUT) for all possible point pairs $(x, y) \in Z$ as follows. The maximum Euclidean distance between the reference point $\mathbf{p}_0$ and any fragment point $\mathbf{b}_i$ is bounded by the fragment length N. Hence, the area under the lower right quadrant with center $\mathbf{p}_0$ corresponds to the total number of possible fragment point locations and defines I. Consequently, we can precompute a square matrix of size $|Z| \times |Z|$ holding the angular descriptions according to Equation (3.2) for all tuples $(x, y) \in Z$. This matrix then serves as LUT during feature calculation for arbitrary fragment points.

3.2.3 Discriminative Fragment Learning

For discriminatively learning our proposed contour fragment descriptor, we use the Hough Forest as described in the previous Chapter. Each tree is constructed on a set of training samples, where the input domain $\mathcal{X}$ is represented by pairs $(\mathbf{u}, \Lambda) \in \mathbb{Z}^2 \times \mathbb{R}^{N \times N}$, i.e. each input sample consists of the fragment descriptor matrix Λ associated with the fragment center $\mathbf{u}$. The output space $\mathcal{Y}$ is, as already discussed in Section 2.6.3 of this Thesis, defined by a set of pairs $(c_l, \mathbf{d})$, where $c_l \in \{0, 1\}$ is the class label and $\mathbf{d}$ is the offset vector, describing the displacement to the object center hypothesis.

Once the entire Forest is constructed, the detection process can be started on the test images. Edges are extracted from the test images and arranged into ordered, connected edge lists B_i with $|B_i| \geq N$. For each B_i, again a total number of $(|B_i| - N + 1)$ fragment descriptors $\{\mathbf{u}_j, \Lambda_j\}$ are computed and then classified into tree-specific leaf nodes. Please note that the descriptors are only computed along edges, which significantly reduces the computational costs in comparison to a sliding window approach or dense sampling.

3.2.4 Ranking and Verification

The previous stages of our method provide object hypotheses in the test image and a corresponding score obtained from the Hough votes. Similar to related work [Ommer and Malik, 2009, Maji and Malik, 2009, Riemenschneider et al., 2010] we additionally provide a ranking according to a pyramid matching kernel (PMK) [Grauman and Darrell, 2005] where histograms of oriented gradients (HOG) are used as features. The PMK classifier is trained on the same training examples as used for the Hough forest. We use the classifier for ranking and for verification where we additionally consider nearby locations and scales around the proposed hypotheses. Including the local search is still efficient since our hypotheses generation stage delivers only a few hypotheses per image, therefore an order of magnitude fewer candidates have to be considered as in a sliding window approach.

3.3 Experimental Evaluation

In order to demonstrate the quality of our proposed method, we performed several experiments: First, we demonstrate improved performance of our novel fragment descriptor in comparison to several related shape descriptors as shown in Section 3.3.1. Subsequently, we show the performance of our method on standard benchmark data bases like the ETHZ shape data base [Ferrari et al., 2006] and the INRIA horses data base [Jurie and Schmid, 2004] (Section 3.3.2). We compare our results to several contour-based recognition approaches while we deliberately ignore methods additionally using segmentations or appearance information as in [Toshev et al., 2010, Li et al., 2010].

3.3.1 Fragment Descriptor Evaluation

Due to the large variety of shape-based descriptors, it is vital to evaluate our proposed descriptor in a quantitative experiment. Therefore, we designed an experiment to evaluate different shape descriptors within different learning algorithms for a classification task. In particular, we compared our proposed descriptor to five types of descriptors, namely the chord angle (CA) [Donoser et al., 2009], beam angle (BA) [Payet and Todorovic, 2010], partial contours (PC) [Riemenschneider et al., 2010], turning angle (TA) [Chen et al., 2008] and shape context (SC) [Belongie et al., 2002]. The learning algorithms we used were random forest, linear SVM and a simple nearest neighbor classifier.

We define a test setup where we classify individual edge pixels on Berkeley edges, detected in the giraffe category test images of the ETHZ shape data base. Specifically, we compare the per-pixel classification results to the ground-truth edge annotations. Hence,

Fragment length	l=51	l=41	l=31
Chord Angle (CA) [Donoser et al., 2009]	41.67	43.15	43.43
Beam Angle (BA) [Payet and Todorovic, 2010]	41.61	42.67	43.48
Partial Contours (PC) [Riemenschneider et al., 2010]	41.13	42.68	42.99
Turning Angle (TA) [Chen et al., 2008]	40.49	42.14	42.81
Shape Context (SC) [Belongie et al., 2002]	37.13	37.51	**38.33**
Proposed method	**33.60**	**35.93**	38.58

Table 3.1: Evaluation of descriptor performances at several lengths, showing the per-pixel classification error in % (see text for definition) for each descriptor when learned in a random forest. Our descriptor yields to the classification error of 33.60% for a length of 51, lower than all compared shape descriptors.

we define a classification error, describing the ratio of false classifications to all edge pixels. The protocol for the experiment uses 50% of the images for learning a classifier and 50% for testing. We extracted fragments (or quadratic patches containing edges for SC) at varying sizes from the images, such that for all images a reasonable number of foreground edges remained in the test set.

In Table 3.1 we list some selected results for fragment lengths / patch sizes of 31, 41 and 51 pixels, when learning the individual descriptors in a random forest framework. As shown, our proposed descriptor outperforms all of the other descriptors at length $l = 51$ and is hence well suited for use in a discriminative setting. Please note that increasing the length even more may result in better classification scores, however, the number of edges belonging to the object category contour might decrease. Using linear SVM or nearest neighbors for classification results in approximately similar distributions of the scores. However, the mean error is on average about $5 - 10\%$ higher which suggests that random forest like classifiers are better suited for our task.

3.3.2 Object Detection

Object detection performance is evaluated on the ETHZ shape data base [Ferrari et al., 2006], the INRIA horses data base [Jurie and Schmid, 2004] and for serially sectioned paper fiber specimen. For all our experiments, we use the following parameters. Our forest consists of 12 Hough trees, each with a maximum depth of 15.

The fragment length is fixed to $N = 51$, as suggested from the classification task in the previous section. We randomly extract 10 000 positive training samples from edges within the bounding box annotation. The positive training images are all scaled to the median height of the selected training data base and the aspect ratio is fixed. 10 000 negative training samples are extracted from the same training images, but outside of the bounding boxes. Detection performance is evaluated using the PASCAL 50% criterion.

Since our method is not implicitly scale invariant, we run the detector on multiple scales. However, this can be done efficiently since descriptor calculation takes constant time due to the use of Look-Up-Tables and traversing the trees has logarithmic complexity.

ETHZ shape data base The ETHZ shape data base consists of five object classes and a total of 255 images. The images contain at least one and sometimes multiple instances of a class and have a large amount of background clutter. All classes contain significant intra-class variations and scale changes. Therefore, we run the detector on 15 different scales, equally distributed between factors of 0.2 and 1.6. We use the same test protocol as specified in [Ferrari et al., 2009] where a class model is learned by training on half of the positive examples from a class, while testing is done on all remaining images from the entire data base.

In Table 3.2 and Table 3.3 we list the results of our described object detector for each object class in comparison to [Ommer and Malik, 2009, Maji and Malik, 2009, Riemenschneider et al., 2010, Yarlagadda et al., 2010] where divisions into voting, ranking and verification stages are applicable. However, due to the large number of competing methods, we only provide the scores of the initial Hough *voting* stage and the PMK *ranking* stage in tabular form. Recognition performance is evaluated by ranking the hypotheses according to their confidence scores. In the initial voting stage this confidence corresponds to the accumulated values in the Hough space. For ranking, the confidence scores are updated using the HOG-based verification as described in Section 3.2.4. Both *voting* and *ranking* are evaluated at 1.0 FPPI. Ranking is quite efficient since on average only 3.5(!) hypotheses are returned by our method. Finally, we also show results of our method for the full verification step (where also nearby locations and scales are tested around the returned hypotheses) at FPPI = 0.3/0.4, which is the standard measure for comparing results on ETHZ data base.

As can be seen in Table 3.2 and Table 3.3, we outperform comparable methods after both, Hough-voting and ranking stage. We achieve a performance boost of **7.5%** over [Yarlagadda et al., 2010], 18.1% over [Riemenschneider et al., 2010], 26.6% over [Ommer and Malik, 2009], 24.5% over [Maji and Malik, 2009] and even 34.2% over [Ferrari et al., 2009]. After applying the learned HOG models for ranking we are 11.0% better than [Riemenschneider et al., 2010] and 15.6% better than [Ommer and Malik, 2009] ([Yarlagadda et al., 2010] has no scores for ranking). Finally, our method also performs well with respect to the full verification system, providing an average recognition score of 93.3/96.1 at 0.3/0.4 FPPI,

			Voting Stage (FPPI=1.0)				
ETHZ	Hough	M^2HT	w_{ac}	PC	Group	Hough Forest	**Our**
Classes	[Ferrari et al., 2009]	[Maji and Malik, 2009]	[Ommer and Malik, 2009]	[Riemenschneider et al., 2010]	[Yarlagadda et al., 2010]	[Gall and Lempitsky, 2009]	**work**
Apples	43.0	80.0	85.0	90.4	84.0	80.0	94.4
Bottles	64.4	92.4	67.0	84.4	93.1	70.8	90.9
Giraffes	52.2	36.2	55.0	50.0	79.5	60.5	86.7
Mugs	45.1	47.5	55.0	32.3	67.0	73.1	92.3
Swans	62.0	58.8	42.5	90.1	76.6	81.3	73.3
Average	53.3	63.0	60.9	69.4	80.0	73.1	**87.5**

Table 3.2: Hypothesis voting stage at 1.0 FPPI (using PASCAL50 criterion) for the ETHZ shape data base [Ferrari et al., 2006]. Our coverage score increases the performance by **7.5%** [Yarlagadda et al., 2010], 18.1% [Riemenschneider et al., 2010], 24.5% [Maji and Malik, 2009], 26.6% [Ommer and Malik, 2009] and 34.2% [Ferrari et al., 2009].

which is approximately on par with scores reported from contour-based approaches in [Srinivasan et al., 2010] (95.2/95.6). Please note however that their method has higher computational complexity and detection takes minutes per image.

Another experiment compares our proposed descriptor with respect to the Hough Forest learning environment, we trained and evaluated on the same number of trees using appearance-based image features as available in the implementation of [Gall and Lempitsky, 2009]. To have a fair comparison and accomplish each of the 15 considered scales in reasonable time, we evaluated on the same locations as we did for our descriptor. As shown in Table 3.2, our single feature descriptor clearly outperforms the standard Hough Forest using all 32 features (14.4% better).

We are aware and acknowledge higher scores reported in [Toshev et al., 2010] (94.3/96.0) and [Li et al., 2010] (average precision of 90.25% at 0.02 FPPI) but also explicitly stress their incorporation of segmentation information in the latter cases.

In Figure 3.5 we show the detection rate vs. FPPI plots for all ETHZ classes in comparison to all results provided by Ferrari *et al.* [Ferrari et al., 2008, Ferrari et al., 2009]. Figure 3.6 illustrates some results for different classes. We show the cluttered edge responses of the test images used for localization and the corresponding reprojections of the classified fragments for an object hypothesis. In addition, they can be used to approximately delineate the object contour. The inpainted circles indicate the voting centers. The degree of intensity along the edges corresponds to the weights of the extracted fragment around this point, where darker is more important.

INRIA horses data base The INRIA horses data base [Jurie and Schmid, 2004] contains a total number of 340 images where 170 images belong to the positive class showing at least one horse in side-view at several scales and 170 images without horses. The experimental setup is chosen as in [Ferrari et al., 2009] where the first 50 positive examples are used for training and the rest of the images are used for evaluation

	Ranking Stage (FPPI=1.0)			*Verification Stage (FPPI=0.3/0.4)*
ETHZ Classes	[Ommer and Malik, 2009] + PMK	[Riemenschneider et al., 2010] + PMK	**Ours** **+ PMK**	**Our work** **Verification**
Apples	80.0	90.4	100.0	94.4/100
Bottles	89.3	96.4	95.5	100/100
Giraffes	80.9	78.8	93.3	91.1/93.3
Mugs	74.2	61.4	88.5	80.8/87.2
Swans	68.6	88.6	93.3	100/100
Average	78.6	83.2	**94.2**	**93.3/96.1**

Table 3.3: Ranking and verification stages showing detection rates (using PASCAL50 criterion) for the ETHZ shape data base. After ranking we achieve an improvement of **11.0%** over [Riemenschneider et al., 2010] and 15.6% over [Ommer and Malik, 2009].

(120 + 170). We run the detector on 8 different scale factors between 0.5 and 1.5. We achieve a competitive detection performance of **85.50%** at 1.0 FPPI, compared to presented scores in [Maji and Malik, 2009] (85.3%) and [Yarlagadda et al., 2010] (87.3%). Please note that [Maji and Malik, 2009] additionally includes different aspect ratios in the voting stage which is not the case in our system. Moreover, we outperform the methods in [Riemenschneider et al., 2010] (83.72%), [Ferrari et al., 2008] (80.77%) and [Ferrari et al., 2007] (73.75%).

Paper fiber cross section detection In the next experiment we evaluate the detection of serially sectioned paper fiber specimen. Analyzing the cellulose fiber network in a sheet of paper is of broad interest in academia and paper industry, because mechanical paper material properties such as strength and stiffness are highly determined by the fiber network and fiber-to-fiber bondings. Conventional paper basically consists of a network of thousands of paper fibers, embedded in a filler medium and enclosed by a so-called coating layer. The individual fibers follow no designated pattern when changing their morphology. Additionally, a lack of significant appearance information makes edges the only reliable cue for detection.

We evaluated our proposed method on microtomy images [Wiltsche et al., 2005] of eucalyptus paper samples. The data set consists of three image sequences, where each sequence contains 20 images. Paper technology experts manually segmented some fiber cross sections in each image, yielding ground truth data for 7 327 individual fibers (some exemplary shapes are shown in Figure 3.9). Out of this pool, we randomly selected 500 cross sections for training and computed 20 fragment descriptors per edge.

Please note, that the images were not entirely segmented, i.e. not all fibers were extracted, such that there is a certain amount of label noise in the negative training data.

In Table 3.4 we list the detection scores for the three image sequences. We evaluated the scores similar to the approach presented in [Shotton et al., 2005] and consider peak hypotheses in the Hough space (obtained from standard non-maxima suppression) with deviations of ≤ 25 pixels from the ground truth bounding box centers as correct. On the three image sequences, we obtain a promising average detection rate of 81.86%. Since the data sets were not entirely segmented (many fibers were left non-annotated), we can only provide the recall for the annotated ones. Also, we need to rely on qualitative examination (see Figure 3.10), due to high efforts needed for the annotation procedure.

Sequence #	Average detection in %
1	80.76
2	83.22
3	81.60
Average	81.86

Table 3.4: Detection scores on eucalyptus paper sequences.

Fig. 3.10 shows our detection results on two successive microtomy images containing differently shaped and sized fiber cross sections. The qualities of the respective detection hypotheses in the Hough image are visualized by superimposing the original images with heat maps, where 'red' corresponds to a strong indicator. As can be seen, fibers with strong variations in their shapes are detected with high confidence. The regions of false detections are small which allows to estimate another important quantity in paper science - the local density of fibers in a paper sample.

3.4 Conclusion

In this chapter we investigated the use of contour fragment descriptors in a random forest learner for the task of object detection. Our method discriminatively learns a number of local contour fragment descriptors in combination with their spatial location relative to object centroid in a Hough Forest classifier. We designed a fragment descriptor that abstracts spatially connected edge points into angular relations in a matrix form and demonstrated that our proposed descriptor shows distinctive patterns for differently shaped fragment primitives, while tolerating small perturbations and intra-class variabilities. Experiments demonstrated that the proposed descriptor outperforms related shape descriptors. We further presented convincing results on the well-known ETHZ and INRIA horses data bases. Moreover, we demonstrated the suitability of our method for the task of paper fiber cross section detection. In addition, we demonstrated that back-projections of the fragments voting for a hypothesis allows delineating the object outline.

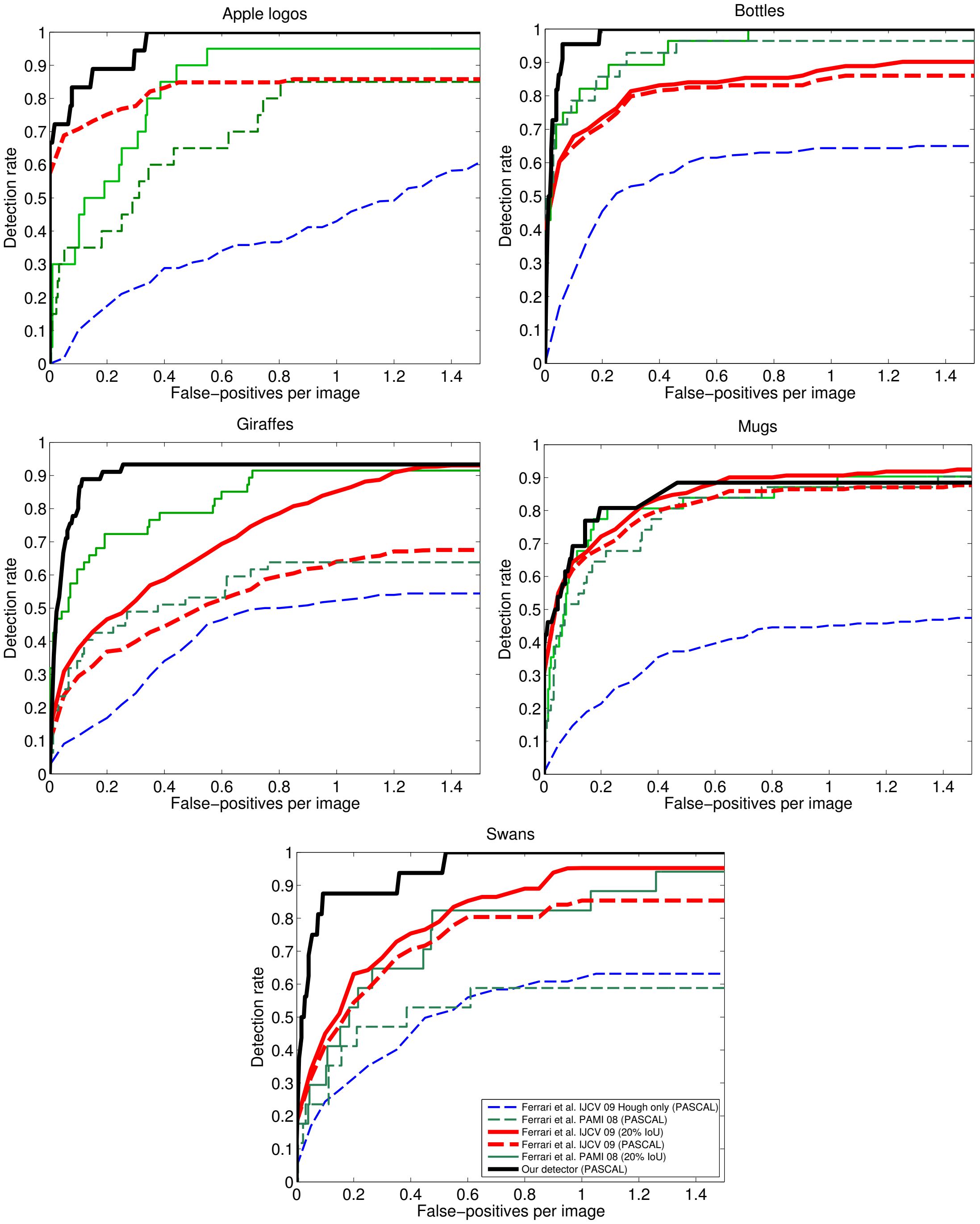


Figure 3.5: Object detection performance on ETHZ in comparison to [Ferrari et al., 2008, Ferrari et al., 2009]. Each plot shows curves for the 50% Pascal criterion and the 20%-IoU criterion of the methods proposed in [Ferrari et al., 2008, Ferrari et al., 2009]. Our results for 50% PASCAL criterion are shown in thick solid black. Note, that we mostly outperform results of [Ferrari et al., 2008, Ferrari et al., 2009] consistently over all classes although evaluating under the stricter 50% PASCAL criterion.

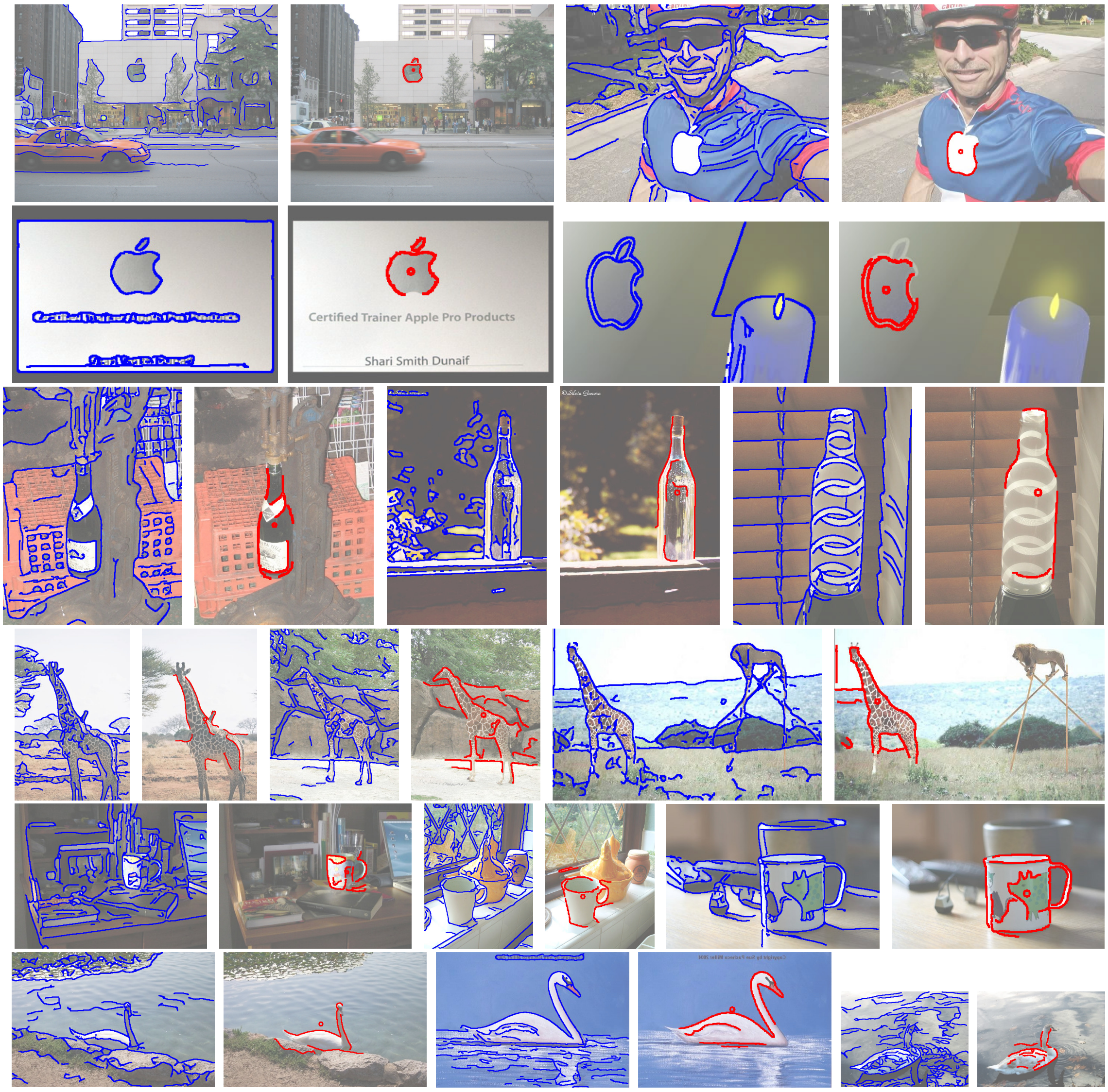


Figure 3.6: Examples for successful object localizations for classes of ETHZ. The cluttered edge responses (in blue) and the reprojected fragments (in red) for the object hypothesis with the highest confidence per image are shown. Best viewed in color.

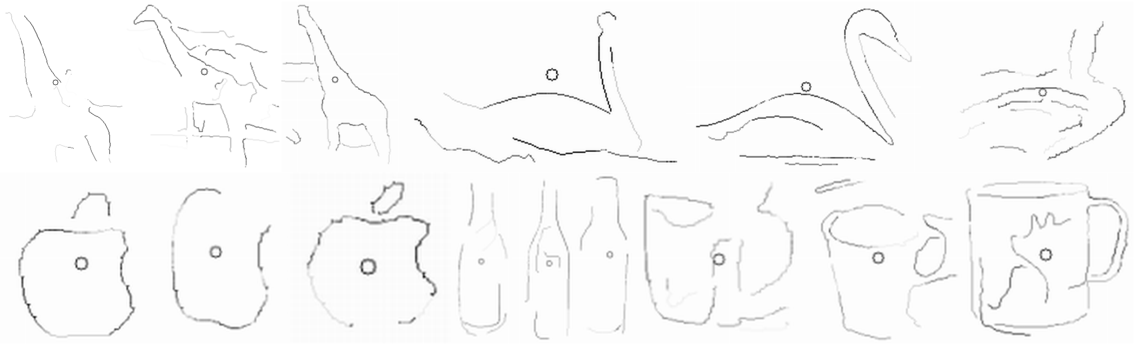


Figure 3.7: Examples of reprojected contour fragments for a detected object hypothesis for all five classes of ETHZ. As can be seen reprojections allow to approximately delineate the object outline.

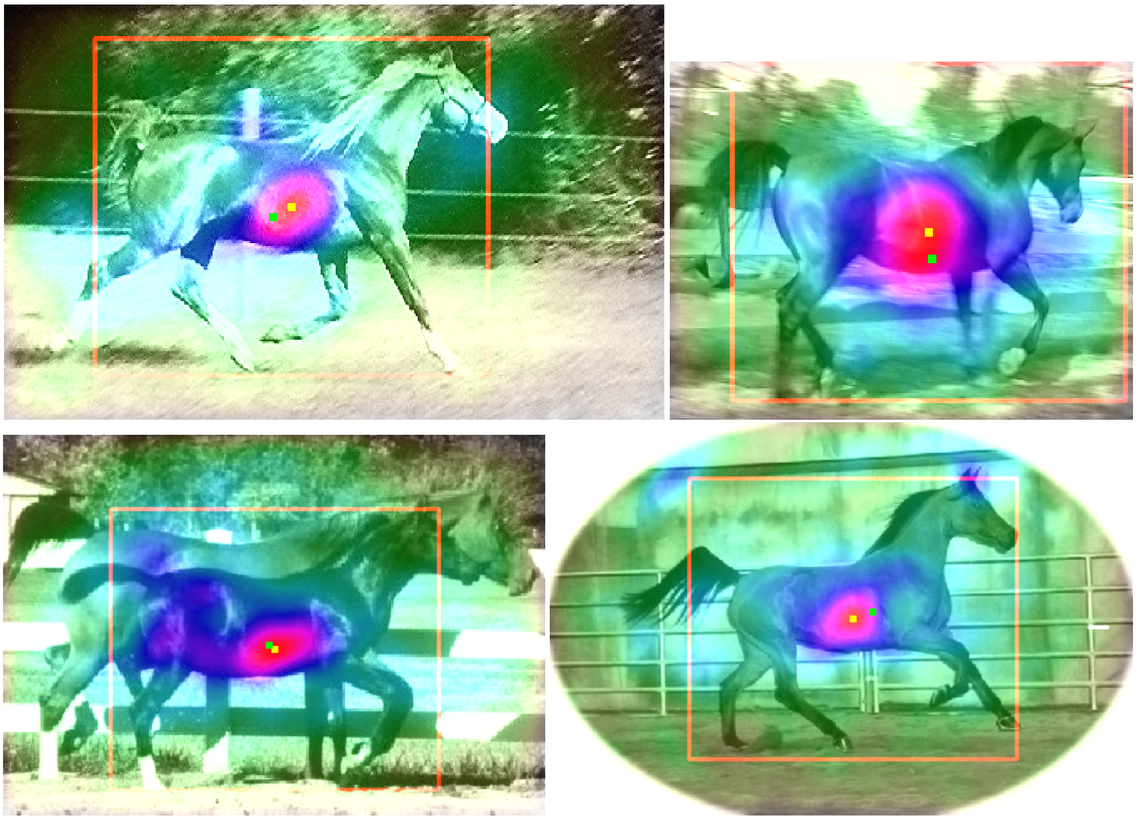


Figure 3.8: Example centroid detections, superimposed on images of INRIA horses data base. Green points denote ground truth centroid, yellow points indicate detected maxima identified by our method. Additionally, the corresponding bounding boxes of the respective scales are provided.

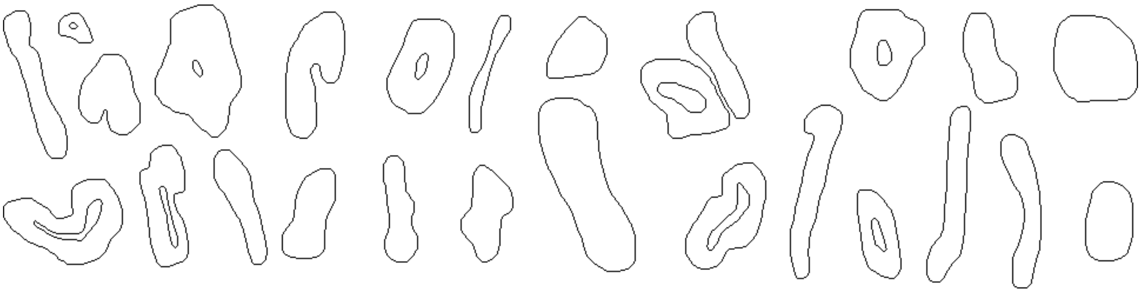


Figure 3.9: Selection of differently shaped fibers we want to detect in microtomy images.

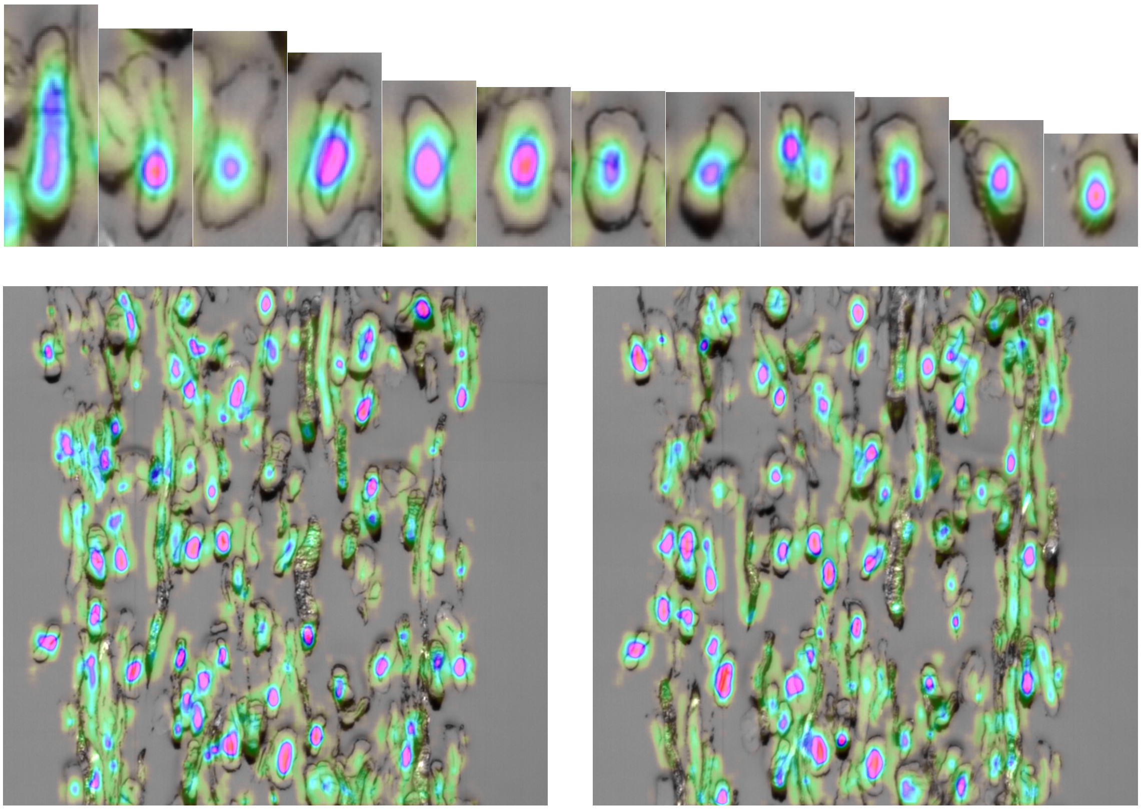


Figure 3.10: Top row: Cropped detections of differently shaped fiber cross sections. Bottom: Microtomy images of two successive eucalyptus paper samples superimposed with heat maps of detections. Best viewed in color.

Chapter 4

Evolutionary Hough Games for Coherent Object Detection

Contents

In this chapter we address the important topic of non-maxima suppression (NMS), i.e. a post-processing step that is typically applied on the output of a given predictor to obtain final object location hypotheses. We propose a game-theoretic approach for finding multiple instances of an object category as sets of mutually coherent votes in a generalized Hough space. In particular, we analyze contextual relations of sample pairs by jointly considering their respective classification and regression predictions, stemming from a Hough forest. Most Hough-voting based detection systems have to apply parameter-sensitive NMSor mode detection techniques for finding object center hypotheses. Moreover, the voting origins contributing to a particular maximum are lost and hence mostly bounding boxes are drawn to indicate the object hypotheses. To overcome these problems, we introduce a two-stage method, applicable on top of any Hough-voting based detection framework. First, we define a Hough environment, where the geometric compatibilities of the voting elements are captured in a pairwise fashion. Then we analyze this environment within a game-theoretic setting, where we model the competition between voting elements as a Darwinian process, driven by their mutual geometric compatibilities. In order to find multiple and possibly overlapping objects, we introduce a new enumeration method inspired by tabu search. As a result, we obtain locations and voting element compositions of each object instance while bypassing the task of NMS. We demonstrate the broad applicability of our method on data bases like the extended TUD pedestrian crossing scene.

4.1 Introduction and Related Work

Due to its generality, the Hough concept has gained much attention in the field of part-based object detection as e.g. in [Leibe et al., 2008, Gall and Lempitsky, 2009, Maji and Malik, 2009, Gall et al., 2011]. Typically, such methods construct a codebook from training data for the desired object category, containing local patch descriptors (voting elements) together with their respective center vote information. Given e.g. bounding box annotated training data, the center vote information is simply obtained as offset vector between the voting element within the bounding box and its respective center. Since the Hough space representation is decoupled from the original image domain, there are no restrictions on how the voting elements have to be obtained. For example, some methods use interest points [Leibe et al., 2008], dense sampling [Gall and Lempitsky, 2009], boundary fragments [Opelt et al., 2006] or edge groups [Yarlagadda et al., 2010]. In the detection step, the test image is matched to the codebook and the corresponding voting elements are projected in the Hough space. Traditionally, the resulting peaks are analyzed with respect to their confidence values and furthermore treated as possible object location hypotheses. In such a way, analysis of the Hough space should simply boil down to identifying true object locations and discriminating them from wrong ones.

Unfortunately, this maxima detection is complicated by several problems arising in practical Hough-voting detection systems. First, projections of voting elements in the Hough space produce scattered and therefore smeared hypotheses such that there are no unique observations of (local) maxima. Second, once the voting elements are cast to the Hough space, their origin is lost due to the accumulative aggregation. Moreover, this implies that re-associations (i.e. back-projections) between particular maxima and their voting origins are questionable since noisy contributions from non-object votes cannot be explained right away. Third, there is no evidence whether an object is present or not since the number of objects in an image cannot be determined, based on the Hough space alone. Of course, maxima in the Hough space should coincide with true object centers but in practice very often all peaks are collected that exceed a certain, application-specific threshold. Finally, for overlapping or strongly occluded objects, maxima analysis in the Hough space is largely undefined. Therefore, Hough space analysis should not be performed as vanilla maxima thresholding but instead take all contributing voting elements *and* their mutual agreement on the object centroid hypothesis into account.

To obtain meaningful object center hypotheses from the Hough space, NMS techniques are typically applied to resolve at least the first of the above mentioned problems. For instance, NMS aims for suppressing all hypotheses (or bounding boxes) within a certain distance and quality with respect to each other. Various NMS techniques were introduced but most of them turn out to be application-specific heuristics with significant effort for parameter tuning. Recently, Barinova et al. [Barinova et al., 2010] came up with a probabilistic interpretation of the Hough transform, reformulated into a facility location problem in order to bypass the task of NMS. Since the resulting maximization problem is NP-hard, optimization was tackled by iteratively applying a greedy algorithm together with a NMS scheme. Their NMS strategy enforced to pick contributing voting elements only within (a) a pre-defined range and (b) voting quality above a certain threshold. As a result, they provided a bounding box delimited object detection.

In this work, we propose a novel method for finding (multiple) instances of an object category, applicable on top of arbitrary Hough-voting based detection frameworks like the Hough Forest [Gall and Lempitsky, 2009] or the Implicit Shape Model (ISM) [Leibe et al., 2008]. More specifically, we detect and describe each object as the set of geometrically coherent voting elements, meaning that each of our final object hypotheses consists of all mutually compatible voting elements. This is in contrast to bounding-box delimited object detections and makes additional NMS unnecessary. Our method mainly consists of the following two steps.

First, we are analyzing the voting element information in a pairwise manner. Therefore, we are introducing a compatibility function that transfers pairs of votes with respect to their hypothesized object center location and their geometrical constellation into a compatibility matrix representation. In such a way, we combine geometric information like orientation and object center agreement between voting elements together in a novel *Hough environment* formulation.

In the second step, we jointly assemble the individual object instances as geometrically coherent sets by introducing a novel, game-theoretic detection approach. Specifically, we model the competition between contributing voting elements as a Darwinian process within the context of *Evolutionary Game Theory* (EGT) [Weibull, 1995], where the mutual geometrical compatibilities drive the selection mechanism. This process imitates the well-known principle of survival of the fittest, which in our case corresponds to identifying the subset of voting elements that best possibly assemble the object category to be found. As a result, we obtain precise information about presence, location and composition of an object at reasonable computational time. The advantages of our method over standard Hough space analysis are subsumed as follows:

- We exploit geometric information provided by the individual voting elements rather than analyzing the accumulative Hough space. Enforcing structural coherence prunes away spurious location hypotheses.

- The proposed evolutionary game theoretical formulation identifies sets of mutually compatible, coherent voting elements with respect to arbitrary, learned object categories.
- The nature of the evolutionary process can leave voting elements unassigned, i.e. spurious contributions, noise and outliers are massively suppressed or even ignored.
- The proposed method is also capable of assigning voting elements to multiple objects. This is particularly helpful when objects occlude each other.

A disadvantage of the proposed method (in its general formulation) can be found in the quadratically growing compatibility matrix, holding the pairwise voting information. However, we show ways to incorporate prior knowledge about the object category from training data as well as sampling strategies to overcome this limitation and to keep the matrix at a reasonable size.

The rest of this chapter is organized as follows. In Section 4.2, we introduce the Hough Environment, i.e. the space where our pairwise, geometric voting compatibilities live. In Section 4.3 we explain some basics of EGT to understand what happens in the biologically inspired voting selection process, before we show our definition of *Hough Games* and our novel detection algorithm. Broad applicability and experimental results of our method are demonstrated in Section 4.4 before we conclude in Section 4.5.

4.2 From Hough Space to Hough Environment

In this section we first introduce the additional notation, necessary in this chapter. Then, we briefly review a probabilistic interpretation of the classical Hough transform, mainly from [Barinova et al., 2010]. Finally, we describe our proposed Hough environment where we include geometric information in a pairwise setting to incorporate a more holistic view in the detection process.

We denote with S the set of N observations (voting elements) from an image. Each observation $i \in S$ has a spatial origin $\mathbf{y}_i$ in the image space, stemming from voting elements and their respective descriptors I_i. We furthermore assume that we are given a classification function $h(i)$ for the class label assignment and a probability score $p(h(i)|I_i)$ for each voting element $i \in S$. In addition, each voting element $i \in S$ obtains a voting vector $\mathbf{d}_i$ after being classified, pointing towards its associated object center (see Figure 4.1). All of the above parameters can be obtained from previous works like the ISM [Leibe et al., 2008] or the Hough Forest [Gall and Lempitsky, 2009, Gall et al., 2011].

Within the generalized Hough transform [Ballard, 1981], object detection is formulated by independently considering each voting element $i \in S$ as being generated from some

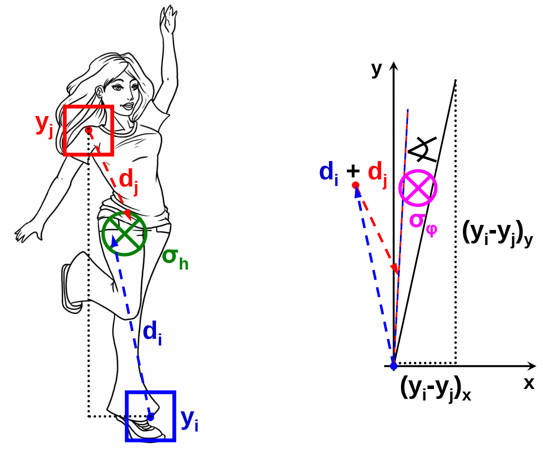


Figure 4.1: Sketches of geometrical features used in proposed Hough environment. The left figure shows the geometry used for voting center compatibility estimation while the right one illustrates the features used to compute orientation compatibility.

object $h \in \mathcal{H}$, where $\mathcal{H}$ is the Hough space, or from no object at all. Let $\mathcal{H}' = \mathcal{H} \cup \{0\}$ be the Hough space augmented with a special element $\{0\}$ and let $x_i \in \mathcal{H}'$ be a random variable indicating whether the voting element $i \in S$ has been generated by object $h \in \mathcal{H}$ or by no object $\{0\}$. In this way, votes can be considered as pseudo-densities $V(x_i = h|I_i)$ conditioned on their respective descriptor I_i. Summing up the conditional (pseudo)-densities over all observations is then supposed to establish local maxima in the Hough space which can be analyzed for valid hypotheses.

Assuming complete independence between voting elements is clearly a very rough approximation, especially since adjacent locations are likely to belong to the same object in the image. To bypass this per-sample independence assumption, we propose a new interpretation of the Hough space, namely the *Hough environment*. Instead of accumulatively combining center voting information in the Hough space, we aim for a joint combination with additional pairwise constraints derived from respective origins and orientations. The additional, geometric information helps to infer meaningful object locations, even in cases where no dedicated peaks in the associated Hough space are available. For example, for the task of pedestrian detection in crowded scenes it might suffice to identify votes stemming from the feet or the head-torso combination alone when not all regions of the human bodies are visible due to occlusion. Consequently, the independence assumption is now assumed to hold for pairs of voting elements, providing a more discriminative nature as opposed to single elements. However, pairwise analysis comes at the cost of increased complexity.

In our proposed Hough environment, we explicitly stress the importance of geometry and spatial relations among intra-object voting elements. Therefore, we are modeling joint densities of pairs of voting elements $i, j \in S$ with respect to their agreements on their mutually hypothesized center location c_{ij} in the Hough domain and their relative angular orientation φ_{ij} and in the image domain. See Figure 4.1 for respective illustrations. Please note that our setup is not restricted to the above mentioned features but may also be extended with context or object specific configuration information [Ren et al., 2005, Leordeanu et al., 2007].

The pairwise compatibility for the object center certainty is modelled as a function weighting the distance between the hypothesized centers of voting elements $i, j \in S$ according to

$$p_{c_{ij}} = \exp\left(-\frac{||(\mathbf{y}_i + \mathbf{d}_i) - (\mathbf{y}_j + \mathbf{d}_j)||^2}{\sigma_h^2}\right), \tag{4.1}$$

where σ_h is a parameter to control the allowed deviation. This term may also be considered as pairwise breakdown of the original Hough center projection.

The second component in the Hough environment models the orientational similarities between the considered pair of votes and the actual relative orientation between the spatial origins in the image domain. Hence, we define

$$p_{\varphi_{ij}} = \exp\left(-\frac{\sphericalangle(\hat{\mathbf{y}}_{ij}, \hat{\mathbf{d}}_{ij})^2}{\sigma_\varphi^2}\right), \tag{4.2}$$

where $\sphericalangle(\cdot, \cdot)$ returns the enclosed angle between the normalized vectors $\hat{\mathbf{y}}_{ij} = \frac{\mathbf{y}_i - \mathbf{y}_j}{||\mathbf{y}_i - \mathbf{y}_j||}$ and $\hat{\mathbf{d}}_{ij} = \frac{\mathbf{d}_i + \mathbf{d}_j}{||\mathbf{d}_i + \mathbf{d}_j||}$, mapped on the interval $[0, \pi]$ (see Fig. 4.1, right). This orientation feature penalizes differences between the observed geometric configuration in the image and the provided voting information. σ_φ allows to control the influence of the orientation feature.

By combining the terms in Equ. (4.1) and (4.2), we construct a *compatibility function* $C : S \times S \rightarrow [0, 1]$ defined as follows:

$$C(i, j) = p(h(i)|I_i)p(h(j)|I_j)p_{c_{ij}}p_{\varphi_{ij}}\,. \tag{4.3}$$

Please note that a voting pair (i, j) has to satisfy not only the geometrical constraints formulated in Equ. (4.1) and (4.2) but also needs to be classified as part of the object in order to receive a non-zero compatibility value.

4.3 Non-cooperative Hough Games

Objects within the Hough environment are identified as sets of voting elements exhibiting high mutual compatibilities. However, the characterization and the retrieval of those sets is not a trivial task. Indeed, two objects may have, in general, some voting elements in common (i.e., they may overlap) and, moreover, many voting elements are noisy, i.e., they do not contribute to any object, and should thus be avoided.

Despite this challenging scenario, it turns out that game theory can provide an intriguing and effective solution to our problem. Indeed, we will show in the next section how mutually compatible sets of voting elements can be characterized in terms of equilibria of what we call a Hough game. We will then also cope with the algorithmic problem of extracting those equilibria in order to successfully detect multiple objects from a scene.

4.3.1 Game-theoretic background

Inspired by [Torsello et al., 2006], we define a *Hough game*, i.e., a non-cooperative two-player symmetric game $\Gamma = (S, A)$ [Weibull, 1995], where S is a finite set of *pure strategies* (in the language of game-theory) available to the players, and $A : S \times S \to \mathbb{R}$ is a payoff (or utility) function, where $a_{ij} = A(i, j)$ gives the payoff that a player gains when playing strategy $i \in S$ against an opponent playing strategy $j \in S$. Note that in the sequel we will treat the payoff function A as a matrix which is indexable using elements in S. Within the context of Hough environments, the strategy set S coincides with the set of voting elements, whereas the payoff matrix $A = (a_{ij})$ is defined as follows:

$$a_{ij} = \begin{cases} C(i,j) & \text{if } i \neq j \\ -\alpha & \text{if } i = j\,, \end{cases} \tag{4.4}$$

where C is defined in (4.3) and $\alpha \geq 0$ is a penalization constant, whose role will be clarified later.

Two players with complete information about the Hough game play by simultaneously selecting a voting element from the strategy set and, after disclosing their choices, each player receives a reward proportional to the compatibility of the selected element with respect to the one played by the opponent. Since it is in each players' interest to maximize his own payoff and since no player has prior knowledge about what the opponent is going to choose, the best strategy for a player is the selection of voting elements belonging to a common object. Those elements, indeed, exhibit high mutual compatibilities and by selecting them the chance of earning a higher payoff is increased for both players. Hence, a set of voting elements belonging to a common object arises from what is called an equilibrium of the Hough game. Note that the penalization constant α in the payoff matrix has the important role of preventing trivial solutions where the players select exactly the same voting element. Indeed, we are searching for the configuration of voting elements which yields the maximal coherency with respect to a common object centroid, i.e. an object hypothesis in the Hough environment. In what follows, we provide a formal characterization of game-theoretic equilibria and how they serve our purpose for object detection.

A *mixed strategy* (or randomized strategy) $\mathbf{x} \in \Delta_S$ is a probability distribution over the set of pure strategies S, i.e., an element of the *standard simplex* Δ_S (or simply Δ, if not ambiguous), which is defined as

$$\Delta_S = \left\{ \mathbf{x} : S \to [0,1] \, : \, \sum_{i \in S} x_i = 1 \text{ and } x_i \geq 0 \, , \forall i \in S \right\} . \tag{4.5}$$

This models a stochastic playing strategy for a player, where $x_i = x(i)$ denotes the probability that the player will select the voting element $i \in S$. Note that as for the payoff function, we will treat $\mathbf{x} \in \Delta_S$ as a (column) vector, which is indexable by elements in S. The support of a mixed strategy $\mathbf{x} \in \Delta$, denoted by $\sigma(\mathbf{x})$, is the set of elements with non-zero probability, *i. e.*, $\sigma(\mathbf{x}) = \{i \in S \, : \, x_i > 0\}$. The expected payoff received by a player playing mixed strategy $\mathbf{y} \in \Delta$ against an opponent playing mixed strategy $\mathbf{x} \in \Delta$ is

$$\mathbf{y}^\top A \mathbf{x} = \sum_{i,j \in S} a_{ij} y_i x_j \, . \tag{4.6}$$

An important notion in game theory is that of an equilibrium [Weibull, 1995]. A mixed strategy $\mathbf{x} \in \Delta$ is a *Nash equilibrium* if it is best reply to itself, i.e., for all $\mathbf{x} \in \Delta$, $\mathbf{y}^\top A \mathbf{x} \leq \mathbf{x}^\top A \mathbf{x}$. Intuitively, if $\mathbf{x} \in \Delta$ is a Nash equilibrium then there is no incentive for a player to play differently from $\mathbf{x}$. A refinement of the Nash equilibrium pertaining to EGT, which plays a pivotal role in this work, is that of an *Evolutionary Stable Strategy* (ESS), which is a Nash equilibrium robust to evolutionary pressure in an exact sense [Weibull, 1995]. A mixed strategy $\mathbf{x}$ is an ESS if $\mathbf{x}$ is a Nash equilibrium such that for all $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$, $\mathbf{y}^\top A \mathbf{x} = \mathbf{x}^\top A \mathbf{x} \Rightarrow \mathbf{y}^\top A \mathbf{x} > \mathbf{y}^\top A \mathbf{y}$.

4.3.2 Relations to graph theory and optimization theory

Note that the concept of ESS has an interesting graph-theoretic characterization. Indeed, it coincides with the idea of a Dominant Set, which has been introduced in [Pavan and Pelillo, 2007, Torsello et al., 2006] and generalizes the notion of maximal cliques to edge-weighted graphs. Another interpretation can be performed from an optimization theoretic viewpoint. The detection problem as formulated here can be considered as a quadratic optimization problem (QAP). In [Pavan and Pelillo, 2007], the authors established a one-to-one correspondence between the equilibria of a non-cooperative, two-player game as used in our work and the optima of a QAP. However, the interpretation in terms of an optimization problem holds only true if the payoff matrix fulfills certain properties (like symmetry). The game-theoretic interpretation does not impose such constraints, but instead allows arbitrary compatibilities between the pairs. For example, the advantages of using a game-theoretic formulation for computer vision applications have been demonstrated in [Albarelli et al., 2009] for matching, shape retrieval [Yang et al., 2011] and common visual pattern discovery [Liu and Yan, 2010]. In this line, we motivate the use of ESS for identifying sets of mutually coherent voting elements in the Hough space for performing object detection.

4.3.3 Dynamics for Coherent Object Detection

We will now focus on the computational aspects of our framework, i.e. the extraction of the Ess of a Hough game $\Gamma = (S, A)$. To this end, we undertake an evolutionary setting. Consider a large population of non-rational players, each of which plays a pre-assigned strategy from S, and let $\mathbf{x}(t) \in \Delta$ be the distribution of strategies in the population at time t. Randomly, pairs of players are drawn from the population to play the game Γ and a selection mechanism, where the reproductive success is driven by the payoff gathered by the players, changes the distribution of strategies $\mathbf{x}(t)$ in the population over time. This Darwinian process continues until an equilibrium is eventually reached.

Different dynamics modeling this process have been proposed in Egt and may potentially serve our need of finding Ess of the Hough game. The most famous one is the so-called *Replicator Dynamics* [Weibull, 1995], which are given by

$$x_i(t+1) = x_i(t)\frac{[A\mathbf{x}(t)]_i}{\mathbf{x}(t)^\top A\mathbf{x}(t)}\,. \tag{4.7}$$

Figure 4.2 shows the result of running the Replicator Dynamics on a Hough game. Initially, the population is uniformly distributed over the set of voting elements (patch samples) and, as time passes, the evolutionary pressure drives the distribution mass on the target object, namely the car, which in turn represents an equilibrium of the game. In this work however, we will make use of a new class of dynamics called *Infection and Immunization Dynamics* (InImDyn), which has recently been developed [Rota Bulò and Bomze, 2011, Rota Bulò et al., 2011], and overcomes some limitations of the Replicator Dynamics. The rest of this subsection provides a brief review of InImDyn, refers to the original works for further details and states the limitations of Replicator Dynamics.

The InImDyn dynamics finds an equilibrium of a Hough game $\Gamma = (S, A)$ by iteratively refining an initial mixed strategy $\mathbf{x} \in \Delta$. The procedure is summarized in Algorithm 1. The refinement loop (lines 2–6) continues until $\mathbf{x}$ is close to being a Nash equilibrium according to a tolerance τ, i.e. until $\epsilon(\mathbf{x}) < \tau$, where

$$\epsilon(\mathbf{x}) = \sum_{i \in S} \min\{x_i, (A\mathbf{x})_i - \mathbf{x}^\top A\mathbf{x}\}^2 \tag{4.8}$$

measures the deviation of $\mathbf{x}$ from being a Nash equilibrium and it yields zero if and only if $\mathbf{x}$ is a Nash equilibrium.

If the execution enters the loop then $\mathbf{x}$ is not a Nash equilibrium and hence, there exist strategies $\mathbf{y} \in \Delta$ such that $(\mathbf{y} - \mathbf{x})^\top A\mathbf{x} > 0$ (called *infective* strategies according to [Rota Bulò and Bomze, 2011]). In particular the following two infective strategies exist

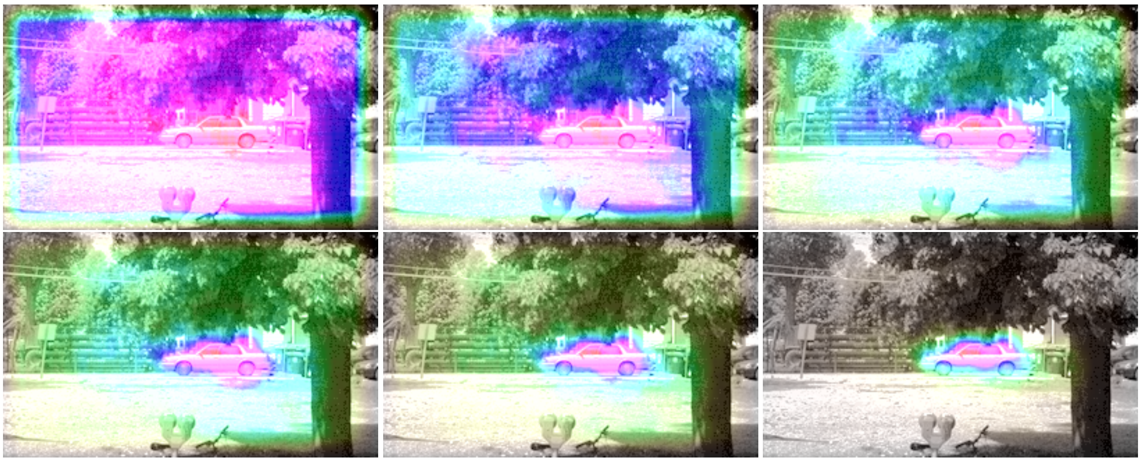


Figure 4.2: Proposed Hough game with its evolutionary process applied to car detection. Starting from homogeneous initialization (top left) until convergence (bottom right) where superimposed, reddish regions correspond to fitness of underlying importance with respect to the sought object geometry. Best viewed in color.

```
Algorithm 1: InImDyn Algorithm for equilibrium selection
   Input: A game $\Gamma = (S, A)$, an initial mixed strategy $\mathbf{x} \in \Delta$ and a tolerance $\tau$
   Output: A Nash equilibrium of $\Gamma$
 1 while $\epsilon(\mathbf{x}) > \tau$ do
 2     $\mathbf{y} \leftarrow \mathcal{S}(\mathbf{x})$
 3     $\delta \leftarrow 1$
 4     if $(\mathbf{y} - \mathbf{x})^\top A(\mathbf{y} - \mathbf{x}) < 0$ then
 5         $\delta \leftarrow \min \left\{ \frac{(\mathbf{x}-\mathbf{y})^\top A\mathbf{x}}{(\mathbf{y}-\mathbf{x})^\top A(\mathbf{y}-\mathbf{x})}, 1 \right\}$
 6     $\mathbf{x} \leftarrow \delta\mathbf{y} + (1 - \delta)\mathbf{x}$
 7 return $\mathbf{x}$
```

[Rota Bulò and Bomze, 2011, Proposition 2]:

$$\mathbf{z}^+ = \mathbf{e}^u \quad \text{and} \quad \mathbf{z}^- = \frac{x_v}{1 - x_v}(\mathbf{x} - \mathbf{e}^v) + \mathbf{x}\,, \tag{4.9}$$

where $u \in \arg\max_{j \in S}(A\mathbf{x})_j$ and $v \in \arg\min_{j \in \sigma(\mathbf{x})}(A\mathbf{x})_j$. Intuitively, in $\mathbf{z}^+$ only the best performing pure strategy (according to a myopic decision) is present in the mixed strategy, while in $\mathbf{z}^-$ the worst one is extinguished. The function $\mathcal{S}(\mathbf{x})$ at line 2 returns the strategy between $\mathbf{z}^+$ and $\mathbf{z}^-$ yielding the largest expected payoff against $\mathbf{x}$, i.e.

$$\mathbf{y} = \mathcal{S}(\mathbf{x}) \in \underset{\mathbf{z} \in \{\mathbf{z}^+, \mathbf{z}^-\}}{\arg\max}\, \mathbf{z}^\top A\mathbf{x}\,. \tag{4.10}$$

Finally, the original strategy $\mathbf{x}$ and the new one $\mathbf{y}$ are linearly combined at line 6 in a way to guarantee a maximum increase in the expected payoff. Indeed, under symmetry assumption of A, the parameter δ computed at lines $3 - 5$ can be regarded to as the solution of

$$\delta \in \arg\min_{\alpha \in \mathbb{R}} \left\{ \mathbf{z}_\alpha^\top A \mathbf{z}_\alpha \, : \, \mathbf{z}_\alpha \in \Delta \right\} , \tag{4.11}$$

where $\mathbf{z}_\alpha = (1 - \alpha)\mathbf{x} + \alpha\mathbf{y}$.

The InImDyn dynamics exhibits desired features in contrast to the Replicator Dynamics. There is a one-to-one correspondence between fixed points of the dynamics and the set of Nash equilibria [Rota Bulò and Bomze, 2011, Theorem 1]. Moreover, there are *strong convergence guarantees* under symmetric payoff matrices [Rota Bulò and Bomze, 2011, Theorems 3,4]: the limit points of the dynamics are Nash equilibria and ESS equilibria are all and the only asymptotically stable points. From a computational perspective, iterations are *linear* in the number of strategies [Rota Bulò et al., 2011, Theorem 8] in contrast to the quadratic ones of the Replicator Dynamics and an equilibrium can be extracted after *finitely* many iterations [Rota Bulò and Bomze, 2011, Theorem 7].

4.3.4 Enumeration of Multiple Objects

Since an evolutionary dynamics will find at most one equilibrium depending on the initial population state, one problem to solve is how to enumerate multiple Ess in order to detect multiple objects. A simple approach consists in starting multiple times the dynamics from random initial points hoping to converge to different equilibria. However, this is an inefficient way of exploring the solution space. Another naive method uses a *peeling-off strategy*, i.e. one can iteratively remove the strategies in the support of newly extracted equilibria from the game. However, this solution cannot find equilibria with overlapping supports and since at each iteration the game is changed, one may potentially introduce equilibria which do not exist in the original game. Due to the lack of a satisfactory solution, we will present a novel heuristic approach for enumerating Ess in this section, which is built upon the novel InImDyn dynamics.

Our enumeration approach is simple and makes use of both InImDyn as an equilibrium selection algorithm and a *strategy tabu-set* in order to heuristically explore the solution space and have a termination guarantee. The algorithm repeatedly interleaves an exploratory and a validation phase. During the exploratory phase, an equilibrium is found by prohibiting the players from using strategies appearing in the tabu-set. By limiting the strategy set, we basically find an equilibrium of a sub-game and not of the original Hough game, but this allows us to explore a new part of the solution space. A validation phase then follows in order to derive an equilibrium of the original game starting from the solution found in the exploratory phase. At each iteration, the tabu-set is increased and this guarantees the termination.

A pseudo-code of our method is shown in Algorithm 2. It takes a game $\Gamma = (S, A)$ as input and returns a set $\mathcal{X}$ of Nash equilibria of Γ. The first two lines of code initialize the solution set $\mathcal{X}$ and our strategy tabu-set $T \subseteq S$ to empty sets (lines 1-2). The algorithm then iterates over the remaining lines of code until the tabu-set equals S. Consider now a generic iteration k over the lines 4-12 and assume $T^{(k)}$ and $\mathcal{X}^{(k)}$ to be the tabu-set and the solution set at iteration k, respectively. In line 4, we create a sub-game $\tilde{\Gamma}^{(k)} = (S \setminus T^{(k)}, A)$, which is obtained from Γ by removing the strategies in the tabu-set $T^{(k)}$ (function SubGame). In line 5, we initialize a mixed strategy $\mathbf{b} \in \Delta_{S\setminus T^{(k)}}$ to the slightly perturbed barycenter of $\Delta_{S\setminus T^{(k)}}$ (function Barycenter), i.e., $b_i \approx 1/|S \setminus T^{(k)}|$ for all $i \in S \setminus T^{(k)}$. We then run InImDyn on $\tilde{\Gamma}$ starting the dynamics from $\mathbf{b}$ and obtain a Nash equilibrium $\mathbf{y} \in \Delta_{S\setminus T^{(k)}}$ of $\tilde{\Gamma}$ (line 6). Note that, differently from a peeling-off strategy, we do not insert $\mathbf{y}$ in our solution set $\mathcal{X}$, because it may not necessarily be an equilibrium of the original game Γ. Instead, we inject $\mathbf{y} \in \Delta_{S\setminus T^{(k)}}$ into Δ_S through the following operator:

$$\pi(\mathbf{y})_i = \begin{cases} y_i & \text{if } i \in S \setminus T^{(k)} \\ 0 & \text{else}\,, \end{cases} \tag{4.12}$$

and use the obtained mixed strategy as starting point of another InImDyn execution, but this time on the original game Γ, from which we obtain an equilibrium of Γ (line 7). We can now safely add $\mathbf{z}$ to $\mathcal{X}$ at line 8 with the only remark that $\mathbf{z}$ may already be in $\mathcal{X}$ (we use the set-union to implicitly remove duplicates). The last step is to update the strategy tabu-set. To this end we distinguish two possible scenarios: 1) the support of $\mathbf{z}$ is contained in the tabu-set, i.e., $\sigma(\mathbf{z}) \subseteq T^{(k)}$, or 2) it has at least one element in $S \setminus T^{(k)}$. In the first case, we insert in the tabu-set all strategies in the support of $\mathbf{y}$. In the second case, we insert in T all strategies in the support of $\mathbf{z}$. [1]

Algorithm 2 will stay in the while-loop for at most $|S|$ iterations as stated by the following proposition. Moreover, at the end, it will return a set of Nash equilibria of Γ.

Proposition 1. *Algorithm 2 terminates within $|S|$ iterations of the while-loop.*

Proof. Since T is initially empty and the while-loop terminates as soon as $T = S$, it suffices to prove that $T^{(k)} \subset T^{(k+1)}$ at each iteration k, where $T^{(k)}$ denotes the content of T at iteration $k \geq 0$ of the while-loop.

Note that $\sigma(\mathbf{y}) \setminus T^{(k)} = \emptyset$ by construction of the game $\tilde{\Gamma}$. Hence, if the algorithm executes line 10 then clearly $T^{(k+1)} = T^{(k)} \cup \sigma(\mathbf{y}) \supset T^{(k)}$, whereas if it reaches line 12 then $\sigma(\mathbf{z}) \setminus T^{(k)} \neq \emptyset$ and therefore $T^{(k+1)} = T^{(k)} \cup \sigma(\mathbf{z}) \supset T^{(k)}$. □

Proposition 2. *Algorithm 2 is correct, i.e., $\mathcal{X}$ contains Nash equilibria of* Γ.

Proof. The proposition holds by noting that we add to $\mathcal{X}$ only equilibira returned by InImDyn when ran on Γ, and the fixed-points of InImDyn are Nash equilibria [Rota Bulò and Bomze, 2011]. □

[1] Note that adding just one vertex from $\sigma(\mathbf{y})$ for the first case, or $\sigma(\mathbf{z}) \setminus T$ for the second one, would be enough as well. This indeed may lead to a better coverage of the solution space at the cost, however, of a larger number of iterations.

Algorithm 2: Nash equilibria Enumeration Algorithm

```
Input: $\Gamma = (S, A)$
Output: $\mathcal{X}$ a set of Nash equilibria of $\Gamma$
$\mathcal{X} \leftarrow \emptyset$
$T \leftarrow \emptyset$
while $T \neq S$ do
    $\tilde{\Gamma} = \textsc{Sub-Game}(\Gamma, S \setminus T)$
    $\mathbf{b} = \textsc{Barycenter}(S \setminus T)$
    $\mathbf{y} \leftarrow \textsc{InImDyn}(\tilde{\Gamma}, \mathbf{b})$
    $\mathbf{z} \leftarrow \textsc{InImDyn}(\Gamma, \pi(\mathbf{y}))$
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{\mathbf{z}\}$
    if $\sigma(\mathbf{z}) \subseteq T$ then
        $T \leftarrow T \cup \sigma(\mathbf{y})$
    else
        $T \leftarrow T \cup \sigma(\mathbf{z})$
return $\mathcal{X}$
```

4.4 Experiments

To demonstrate the quality of our proposed method, we conducted several experiments on both, synthetic and real data sets where we compare to widespread spectral clustering methods. In Section 4.4.1, we provide results for our cluster enumeration algorithm on a synthetically generated data set. Then, we briefly describe how we integrated our Hough game concept in the Hough forest framework [Gall and Lempitsky, 2009] before we show qualitative and quantitative results on real data sets in Section 4.4.3.

4.4.1 Synthetic Data Experiment

First, we demonstrate the capability of robustly enumerating multiple, overlapping clusters under severe amount of noise and outliers. Instead of deriving the payoff matrix as proposed in Equation 4.3, we provide an artificially generated matrix simulating 5 coherent clusters containing different numbers of elements in the range of 140 to 435. The matrix has a size of 1500×1500 and 4 clusters have mutual overlaps in the range of $9 - 60\%$ of their elements. The internal cluster affinities were generated using 70% noise while affinities to non-cluster elements were randomly set with 30% of noise. In addition, we added 300% of uniformly distributed noise on top of the whole matrix. Figure 4.3 shows the detected clusters, forming color-coded block-wise structures. Our method finds the actual clusters with a mean accuracy of 99.33% in terms of F-measure while leaving the rest unassigned. Please note that finding such overlapping clusters is of particular interest since e.g. areas of overlapping objects like persons might be assigned to both individuals.

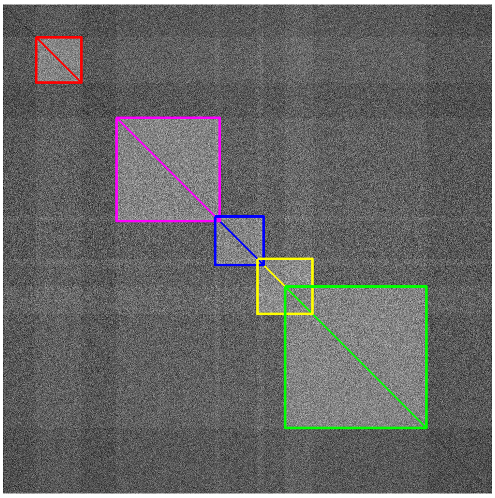


Figure 4.3: Visualization of enumeration results on a synthetically generated payoff matrix containing 5 clusters with varying sizes and overlaps.

4.4.2 Artificial Graph Matching Experiment

In our next experiment we assess the quality of our proposed system by interpreting it as a graph matching problem. Indeed, we can consider the voting element pairs used in the Hough environment as nodes in a graph and their edge weights as a measure for geometric compatibility with respect to a learned reference model (see edge compatibility definitions using Equations (4.1) and (4.2)).

With this experiment we want to stress the independence with respect to the underlying reference model, i.e. our proposed approach can easily be adapted to work with different model generators. Here, the reference graph was constructed from a set of randomly distributed 2D points in a pre-defined region, simulating voting element origins in the image domain. The query graph (i.e. the underlying points we used for the matching task) was formed by independently disturbing the original point locations, followed by randomly rotating and translating the whole point set. In this sense we also know about the ground truth correspondences.

Given the graphs, we define a simple compatibility measure

$$C(ia, jb) = \exp\left(-\frac{D(ia, jb)^2}{2\sigma^2}\right) \text{ using } D(ia, jb) = \frac{|d_{i,j} - d_{a,b}|}{d_{i,j} + d_{a,b}}\,, \tag{4.13}$$

where σ is a normalization parameter and $d_{\cdot,\cdot}$ is the Euclidean distance between point pairs i, j and a, b from the reference and query graphs, respectively.

To demonstrate robustness, we show the noise tolerance of our method by consequently adding outliers to our query graph (drawn uniformly and randomly within the pre-defined region) while analyzing the number of correct matchings. We compare results of our method to two methods for computing the principal eigenvector of C as it is required in spectral methods: power iteration and a variant of the Lanczos method. Each method uses the same difference matrix D as input, analyzing the spatial distances between graph nodes. Please note that the matching quality depends on the normalization of the compatibility matrix and that optimal normalization parameters (here: σ) vary for different optimization approaches. Therefore, we optimized σ for all three methods by exhaustive search over all matching test cases. We found a high variance in the optimal choice for σ, i.e. 5 for our method, 150 for Lanczos and 200 for power iteration.

Figure 4.4 shows average percentage of correct assignments for matching 10 different random reference graphs (each built on 30 points) to query graphs with an increasing number of outliers. As can be seen, all methods can similarly handle a small amount of outliers. However, once the number of outliers exceeds the number of points to match, our proposed algorithm shows improved performance, because of its inherent search for an assignment with maximum internal cohesiveness.

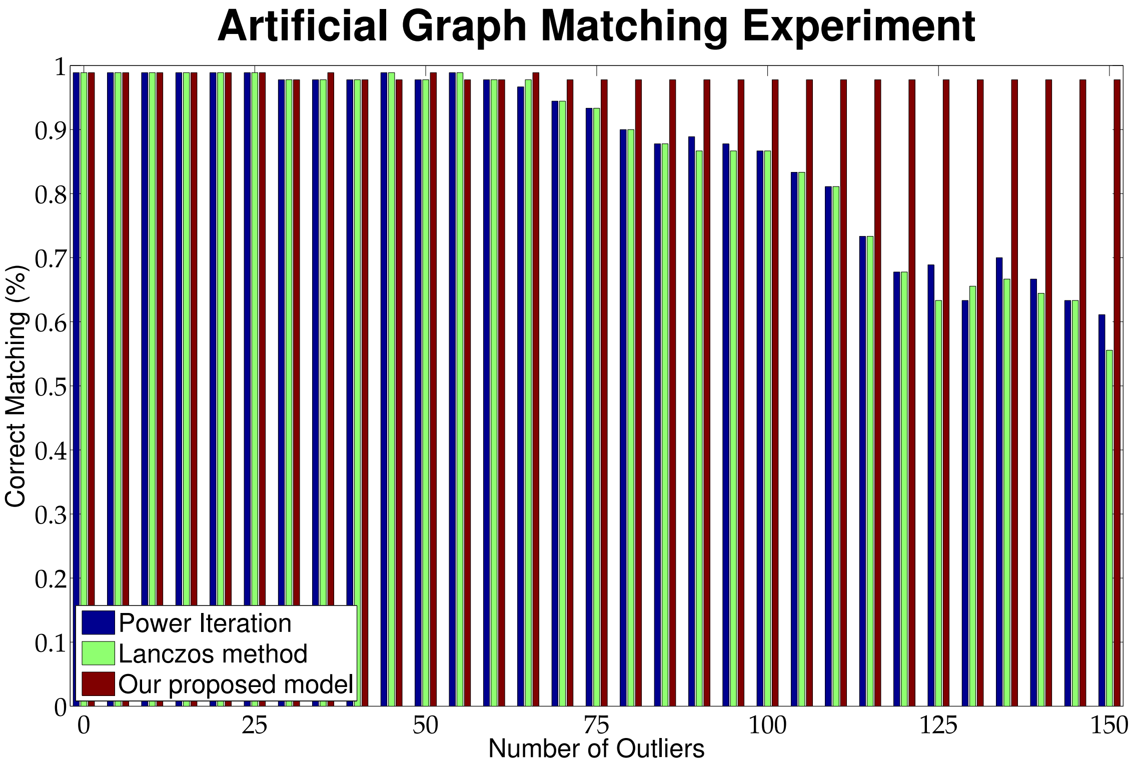


Figure 4.4: Matching reference graphs (30 points) to query graphs with an increasing number of randomly generated outliers (up to 150). The proposed method is compared to two spectral methods: Lanczos and power iteration.

4.4.3 Hough Game Results

We use the Hough forest [Gall and Lempitsky, 2009] for providing the required predictions to construct the Hough environment as outlined in Section 4.2, and we set the parameters $\sigma_h = \sigma_\varphi = 9$. In every Hough tree $t \in \mathcal{T}$, we reduce the set of voting vectors in the leaf nodes to the median vote vector $\mathbf{d}$. Since the Hough Forest provides multiple trees $\mathcal{T}$ for classification, we update Equation (4.3) to $C(i,j) = \sum_{t \in \mathcal{T}} p^t(h(i)|I_i) p^t(h(j)|I_j) p^t_{c_{ij}} p^t_{\varphi_{ij}}$ for computing the compatibilities.

As mentioned at the beginning of this chapter, pairs of voting elements provide better discriminative properties as opposed to single elements, however at the cost of increased computational complexity. Since we want to keep the processed payoff matrices at reasonable size ($< 7k \times 7k$), we constrain the number of considered voting elements to patches with foreground probability ≥ 0.5. Additionally, we consider only pixels lying on a regular lattice with a stride of 2 to reduce the amount of data to be processed. The penalization constant has been empirically determined and was fixed to $\alpha = 10$ for all experiments. Please note that sparse implementation techniques are beneficial, since the pairwise interactions are naturally bounded by the largest possible voting vectors obtained from training data.

Unless otherwise stated, we always grow 15 trees with a maximum depth of 12 on 25 000 positive and negative training samples from the referenced data sets. The considered patch size is 16×16 and all training samples are resized to a similar scale.

We apply our Hough Game for localization of cars on the UIUC cars dataset [Agarwal et al., 2004] and pedestrians on the extended TUD crossing dataset [Barinova et al., 2010]. Additionally, we show qualitative results for enumerating paper fibers in microtomy images, presented in the previous chapter.

UIUC car dataset In our first experiment we evaluate the proposed method on the single scale UIUC car dataset [Agarwal et al., 2004]. The training dataset contains 550 positive and 450 negative images while the test dataset consists of 170 test images showing 210 cars. Although the silhouettes of the cars are mostly rigid, some cars are partially occluded or have low contrast while being located in cluttered background. We achieve a score of 98.5% in terms of equal error rate (EER), hence we are on par with state-of-the-art methods [Lampert et al., 2008, Gall and Lempitsky, 2009] while identifying the set of votes that are corresponding to the individual objects. In Figure 4.5 we show some sample detections and the groups of votes producing the respective detections. Please note how our method is able to deal with partial occlusions and successfully groups coherent object votes.

Extended TUD crossing scene Next, we evaluated on the extended version of the TUD crossing data base [Barinova et al., 2010], showing several strongly occluded pedestrians walking on a cross-walk. The extended version includes also overlapping pedestrians

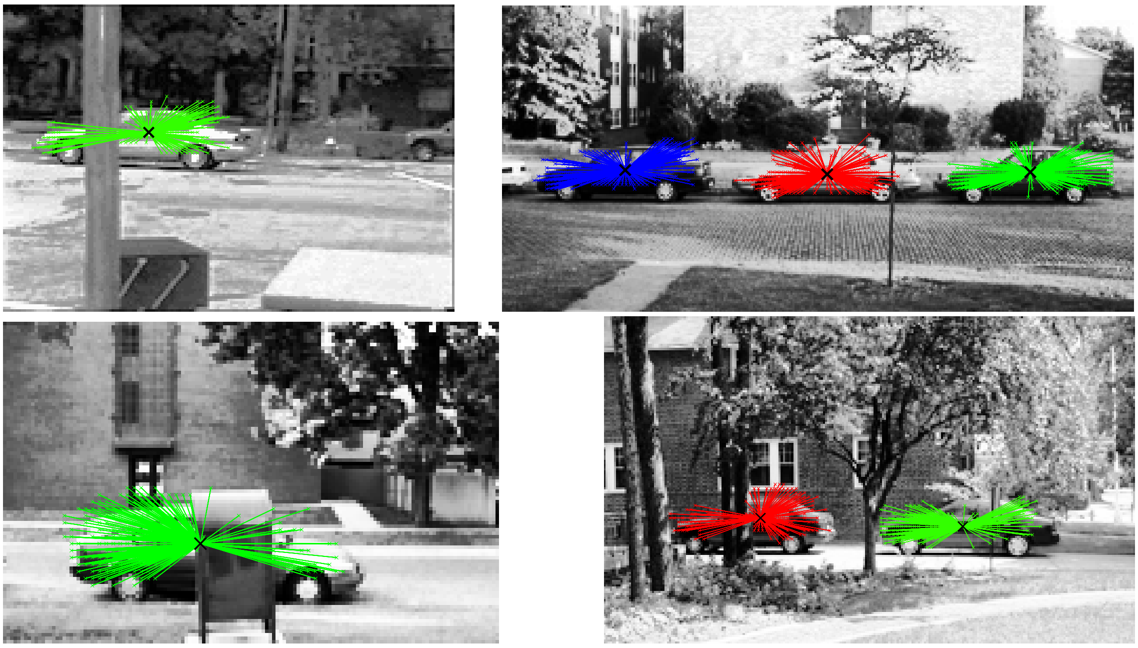


Figure 4.5: Car detections and their contributing votes on selected images of UIUC car databset.

where head and at least one leg are visible. This results in a very challenging data set consisting of 201 images with 1018 bounding boxes. We used the same training protocol as described in [Andriluka et al., 2008]. Since we are not obtaining bounding boxes but rather the sets of contributing voting elements for each person, we decided to evaluate the detection results with the strict criterion introduced in [Shotton et al., 2005]. This criterion accepts detections as correct when the hypothesized center is within 25 pixels of the bounding box centroid on the original scale. In our case, we determined the centroid by taking the median of the reprojected center votes for all detected voting elements. For evaluation, we rescaled the images and the acceptance criterion by a factor of 0.55, such that true positives were counted only within a radius of 13.75 pixels. After constructing the payoff matrices, we played the Hough Game using our novel detection algorithm. To provide a comparison, we handed the same matrices to the widespread normalized cut (nCut) algorithm [Shi and Malik, 2000] and illustrate the results (F-measure per test image) in the top row in Figure 4.6. Since nCut requires the number of clusters to obtain, we evaluated by giving our number of detections as well as giving the ground truth number of persons. As can be seen, our method outperforms the nCut algorithm, even when the true number of objects is provided. We obtain a mean F-measure score of 79.88% compared to 66.56% and 65.23% for nCut provided with ground truth or our detected number of persons, respectively. Since nCut aims for partitioning the whole input into clusters, we tried another setup where we give an additional cluster to the ground truth number and the detected number of persons from our method, respectively. This should allow nCut for partitioning the non-person objects. Before computing the F-measure, we removed the detection associated to the lowest eigenvalue. This resulted in F-measure scores of 61.94% and 58.79%, hence considerably lower than before and suggesting that nCut does not group noise in an individual cluster but rather incorporates it in the individual detections. The bottom row in Figure 4.6 shows color-coded, qualitative results of individual detections of our proposed Hough game. Please note the plausible assemblies of votes from strongly overlapping persons to individual pedestrians, even in the rightmost image, where

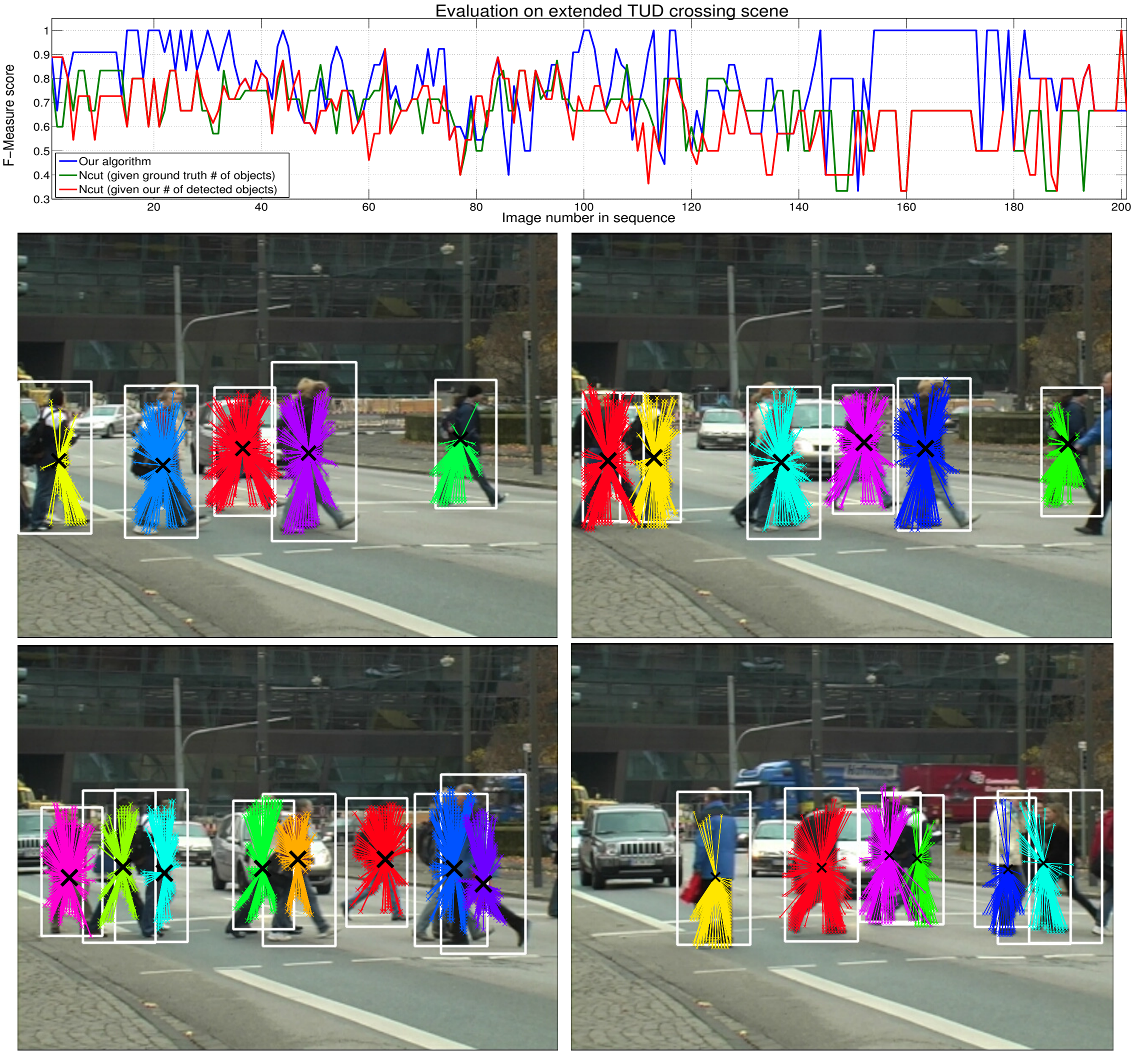


Figure 4.6: Top row: Classification results on extended TUD crossing sequence per image using single scale evaluation. We obtain a mean F-Measure score of 79.88% in comparison to 66.56% and 65.23% for nCut [Shi and Malik, 2000] (provided with ground truth # or our detected # of objects, respectively). Second and third rows: Successive and missing (last image) detections of proposed method. White bounding boxes correspond to ground truth annotations. Best viewed in color.

a person is missed due to assignment of votes to another person in the back. Moreover, it is possible to hypothesize for the person's center by detecting coherent votes of the feet alone (green detection in first image, yellow detection in forth image).

Paper fiber enumeration In this experiment, we apply our method to detect and enumerate individual paper fibers in microtomy images obtained from serially sectioning a paper specimen (see previous Chapter for data set description). In case two (or multiple) fibers are directly next to each other, the fiber walls can be considered as shared features, making their individual detection a well-suited application for the proposed enumeration algorithm. In Figure 4.7 we show qualitative results of fibers in a cross-section image of eucalyptus paper where no ground truth data was available.

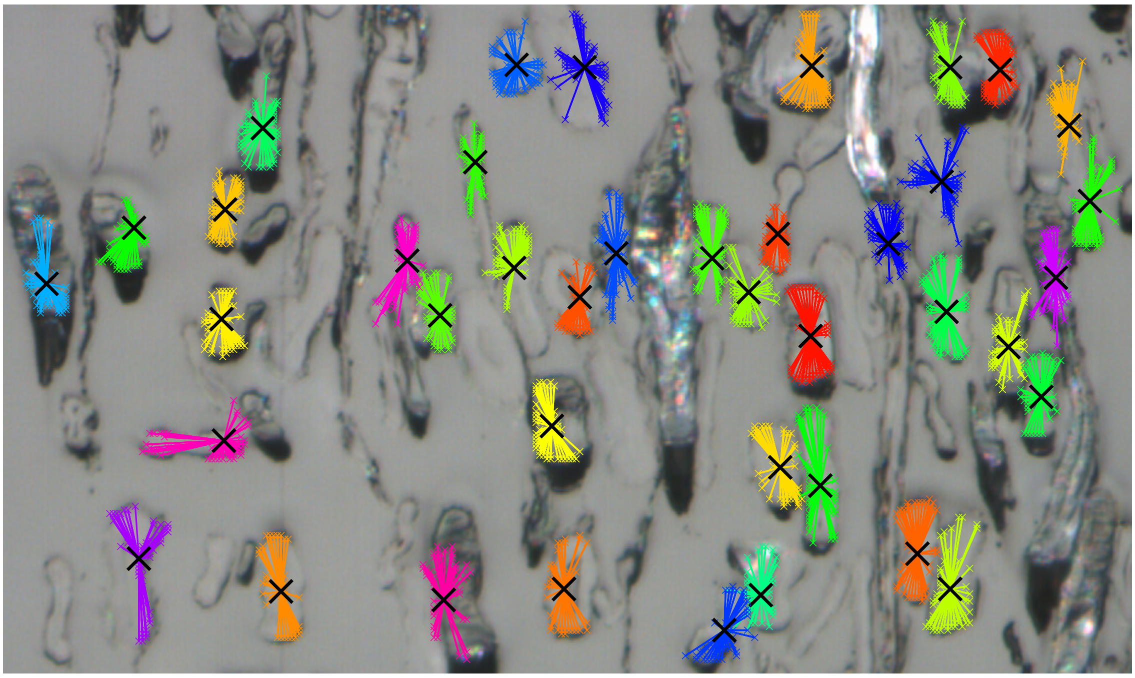


Figure 4.7: Enumerating paper fibers in a cross-section image of eucalyptus paper. Differently shaped fibers were successfully assembled using the proposed enumeration scheme.

4.5 Conclusions

In this chapter we showed a method to identify multiple, possibly overlapping instances of an object category in a Hough-voting based detection framework. We identified individual objects by their sets of mutually compatible voting elements, opposed to analyzing the accumulative Hough Space. We proposed solutions for two challenging problems. First, we introduced the *Hough environment*, where we mapped geometrical compatibilities of votes in terms of mutual object center agreements and orientational similarities. In the second part, we introduced a novel, game-theoretic algorithm for finding multiple objects by assembling pairs of mutually compatible voting elements in the Hough environment. To this end, we designed a *Hough Game* where we modelled the competition of contributing voting elements as Darwinian selection process, driven by geometrical compatibilities with respect to the object category of interest. In contrast to existing methods, our novel detection algorithm can identify several coherent sets with overlapping elements as well as leave non-compatible elements completely unassigned which is particularly helpful in strongly occluded or noisy scenarios. As shown in several experiments, our detection algorithm can successfully cope with severe amounts of noise and outliers.

Chapter 5

Context-Sensitive Decision Forests for Object Detection

Contents

In the previous two chapters, we restricted to *applying* the available Hough forest algorithm either for shape fragment descriptor learning (Chapter 3) or as generator for modelling pairwise compatibilities of test samples with respect to the learned model category (Chapter 4). In this chapter we directly alter the random forest learning algorithm by introducing Context-Sensitive Decision Forests - A new perspective to exploit contextual information in the decision forest framework for the object detection problem. Context-sensitive trees are tree-structured classifiers with the ability to access intermediate prediction information during training and inference time. This intermediate prediction is available to each sample, which allows us to develop context-driven decision criteria, used for refining the prediction process. In addition, we introduce a novel split criterion which in combination with a priority based way of constructing the trees, allows more accurate regression mode selection and hence improves the current context information. In the experiments, we demonstrate improved results for the task of pedestrian detection on the TUD data set when compared to state-of-the-art methods.

5.1 Introduction and Related Work

We are again focusing on the task of object detection, i.e. the problem of localizing multiple instances of a given object class in a test image. While in the previous chapters with have applied Hough forests according to their definition in [Gall and Lempitsky, 2009], we will now extend the underlying learning concept to make them context-sensitive. To this end, we take advantage of the correlations between samples by exploiting their adjacency on the pixel grid as additional feature cues in the learning and inference process. The motivation to do so arises from shortcomings in the prediction process of standard Hough forests, where we typically find non-distinctive object hypotheses in the Hough space, requiring to perform non-maximum suppression (NMS) for obtaining the final results. While this has been addressed in [Barinova et al., 2010] or the previous chapter of this Thesis, another shortcoming is that standard (Hough) forests treat samples in a completely independent way, i.e. there is no mechanism that encourages the classifier to perform consistent predictions with respect to their context.

5.1.1 Early Use of Context in Computer Vision and Pattern Recognition

The importance of context in human visual processes and in our everyday judgments and actions can hardly be stressed enough. It is a common-sense observation, in fact, that objects in the real world do not live in a vacuum, and some thoughts have gone so far as to maintain that all attributions of knowledge are indeed context-sensitive, a view commonly known as contextualism [Cohen, 1986, Price, 2008]. The use of contextual constraints in pattern recognition dates back to the early days of the field, e.g. [Abend et al., 1965] presented an approach for classifying binary random variables, forming patterns on a two-dimensional array. Moreover, at this time context was especially used in connection to optical character recognition problems [Chow, 1962, Toussaint, 1978].

Early efforts to exploit context in computer vision considered contextual dependencies as expert knowledge from a narrow domain, which was provided as a priori information [Yakimovsky and Feldman, 1973, Fischler and Elschlager, 1973, Hanson and Riseman, 1978]. These solutions were however limited due to the inability of handling the uncertainty of real world data. This turned the attention of researchers to developing more general and flexible tools for modeling context. An important pioneering work in this respect appeared in 1976 by Rosenfeld, Hummel and Zucker [Rosenfeld et al., 1976]. They introduced a class of parallel iterative procedures which became a standard technique in the pattern recognition and machine vision domains for many years. These algorithms, generally known as *relaxation labelling processes* [Hummel and Zucker, 1983], attempted to exploit contextual information in order to provide consistent solutions in classification problems where noise and uncertainty can affect the accuracy of classical non-contextual pattern recognition algorithms. Notably, the work in [Pelillo, 1997] provides a formal connection between the heuristics in [Rosenfeld et al., 1976] and the theoretical work in [Hummel and Zucker, 1983].

In the following years, a promising stream of approaches based on Markov Random Fields (MRF) appeared in the computer vision field and led the way to the use of graphical models for modelling contextual relations. Among the most representative works, we mention the one of Geman and Geman [Geman and Geman, 1984], who used a MRF image model for image restoration, and also the book of Stan Li [Li, 1995].

Early works like the aforementioned ones, were focused on exploiting contextual dependences rather than learning them from training samples, an exception being [Pelillo and Refice, 1994], where the problem of learning contextual dependencies within the relaxation labelling framework has been addressed. This is probably due to the limited availability of data and storage capabilities at that time, and the limited computational power. It is nowadays clear to researchers that providing an a priori imprint of contextual relations on their algorithms limits the power of their approaches due to their inability to cope with the variability and complexity of real-world data. Hence, learning the parameters encoding contextual information has become an obliged path to take.

Nowadays, the computer vision community is paying again increasing attention to the role played by contextual information in visual perception, especially in high-level problems such as object recognition or semantic scene segmentation, and neuro-scientists have started understanding how contextual processing takes actually place in the visual cortex. As Bar [Bar, 2004] puts it: ”contextual information can provide more relevant input for the recognition of an object than can its intrinsic properties”, which is something that psychologists working in visual perception know well (see, e.g. [Zusne, 1970]).

5.1.2 Contributions

In this chapter we are proposing that context information can be used to overcome some of the aforementioned problems typically arising in Hough forests. Recently, contextual information has been used in the field of object class segmentation [Rabinovich et al., 2007], however, mostly for high-level reasoning in random field models or to resolve contradicting segmentation results. The introduction of contextual information as additional features in low-level classifiers was initially proposed in the Auto-context [Tu, 2008] and Semantic Texton Forest [Shotton et al., 2008b] models. Auto-context shows a general approach for classifier boosting by iteratively learning from appearance and context information. In this line of research [Montillo et al., 2011] augmented the feature space for an *Entanglement Random Forest* with a classification feature, that is consequently refined by the class posterior distributions according to the progress of the trained subtree. The training pro-

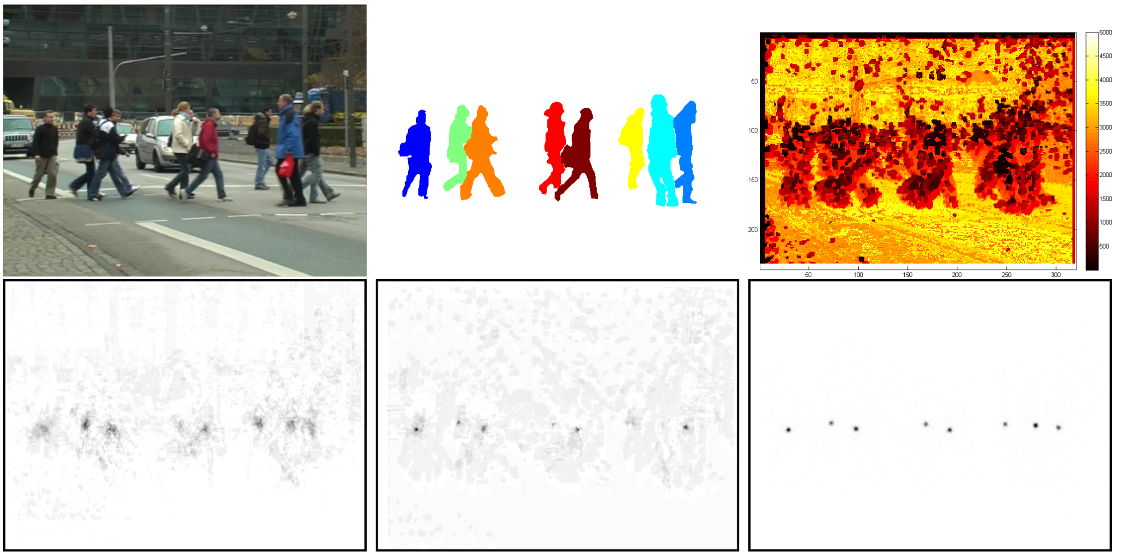


Figure 5.1: Top row: Training image, label image, visualization of priority-based growing of tree (the lower, the earlier the consideration during training.). Bottom row: Inverted Hough image using [Gall and Lempitsky, 2009] and breadth-first training after 6 levels ($2^6 = 64$ nodes), Inverted Hough image after growing 64 nodes using our priority queue, Inverted Hough image using priority queue shows distinctive peaks at the end of training.

cedure is allowed to perform tests for specific, contextual label configurations which was demonstrated to improve the segmentation results.

To this end, we are presenting *Context-Sensitive Decision Forests* - A novel and unified interpretation of Hough Forests in light of contextual sensitivity. Our work is inspired by Auto-Context and Entanglement Forests, but instead of providing only posterior *classification* results from an earlier level of the classifier construction during learning and testing, we additionally provide *regression (voting)* information as it is used in Hough Forests. Another contribution of this work is related to how we grow the trees: Instead of training them in a depth- or breadth-first way, we propose a priority-based construction (which could actually consider depth- or breadth-first as particular cases). The priority is determined by the current training error, i.e. we grow those parts of the tree where we measure the highest training error. To this end, we introduce a unified splitting criterion that estimates the joint error of classification and regression. The consequence of using our priority-based training are illustrated in Figure 5.1: Given the training image with corresponding label image (top row, images 1 and 2), the tree first tries to learn the foreground samples as shown in the color-coded plot (top row, image 3, colors correspond to index number of nodes in the tree). The effects on the intermediate prediction quality are shown in the bottom row for the regression case: The first image shows the regression quality after training a tree with 6 levels ($2^6 = 64$ nodes) in a breadth-first way while the second image shows the progress after growing 64 nodes according to the priority based training. Clearly, the modes for the center hypotheses are more distinctive which in turn yields to more accurate intermediate regression information that can be used for further tree construction. Our third contribution is a new split function that allows to learn from training images containing multiple training instances as shown for the pedestrians in the example. We introduce a test that checks the centroid compatibility for pairs of training samples taken from the context, based on the intermediate classification and regression derived as described before. To assess our contributions, we performed several experiments on the challenging TUD pedestrian data set [Andriluka et al., 2008], yielding a substantial improvement of 9% in the recall at 90% precision rate in comparison to standard Hough Forests, when learning from crowded pedestrian images.

5.2 Context-Sensitive Decision Trees

This section introduces the general idea behind the context-sensitive decision forest without references to specific applications. Only in Section 5.3 we show a particular application to the problem of object detection. Here, the notion of *contextual information* of a sample x is inherited from previous works like [Tu, 2008, Montillo et al., 2011] and it consists of hypotheses about the prediction value that other samples related to x might have at earlier stages of the inference process. As we will demonstrate, the idea of integrating results from previous predictor outputs naturally fits to the hierarchical concept of tree-structured predictors. According to our definition of decision trees in Chapter 2, we can convert each internal node $\textsc{Nd}\,()$ to a leaf node $\textsc{Lf}\,()$, allowing to make a prediction for all data samples it is reached by. Keeping this idea in mind we can make predictions at every growing state of the tree, allowing to establish a notion of context in terms of class labels (classification information) or concentration of voting information (regression information), which in turn can be used when continuing the tree growing process.

A context-sensitive (CS) decision tree is a decision tree in which split functions are enriched with the ability of testing contextual information of a sample before taking a decision about where to route it. We generate contextual information at each node of a decision tree by exploiting a truncated version of the same tree as a predictor. This idea is shared with [Montillo et al., 2011], however, we introduce some novelties by tackling both, classification and regression problems in a joint manner and by leaving a wider flexibility in the tree truncation procedure. For reasons of convenience, we denote the set of CS decision trees as $\mathcal{T}$. The main differences characterizing a CS decision tree $t \in \mathcal{T}$ compared with a standard decision tree are the following:

- *Every* node (leaves and internal nodes) of t has an associated probability distribution $Q \in \mathbb{P}(\mathcal{Y})$ representing the posterior probability of an element in $\mathcal{Y}$ given any data sample reaching it.

- Internal nodes are indexed with distinct natural numbers $n \in \mathbb{N}$ in such a way as to preserve the property that children nodes have a larger index compared to their parent node.

- The split function at each internal node, denoted by $\varphi(\cdot|t') : \mathcal{X} \to \{0,1\}$, is bound to a CS decision tree $t' \in \mathcal{T}$, which is a truncated version of t and can be used to compute contextual information.

Similar to Section 2.2 we denote by $\textsc{Lf}\,(Q) \in \mathcal{T}$ the simplest CS decision tree consisting of a single leaf node parametrized by the distribution Q, while we denote by $\textsc{Nd}\,(n, Q, \varphi, t_l, t_r) \in \mathcal{T}$, the rest of the trees consisting of a node having a left and a right sub-tree, denoted by $t_l, t_r \in \mathcal{T}$ respectively, and being parametrized by the index n, a probability distribution Q and the split function φ as described above.

Different from standard decision trees, we explicitly induce a total ordering on the non-terminal nodes of a specific tree by indexing them with a natural number $n \in \mathbb{N}$. The indexing must satisfy the property that non-terminal children nodes have a larger index as opposed to their parent node (see Figure 5.2). Additionally, also non-terminal nodes have an associated probability distribution $Q \in \mathbb{P}(\mathcal{Y})$ reflecting the posterior probability of elements in $\mathcal{Y}$ given any sample reaching it. Hence, besides leaves, a CS decision tree is a node $\textsc{Nd}\,(n, y, \varphi, t_l, t_r)$, where $n \in \mathbb{N}$ and $y \in \mathcal{Y}$ are the aforementioned index and output value, respectively, φ is a split function that will be described later on and $t_l, t_r \in \mathcal{T}$ are the left and right children of the node.

The split function that comes along with all non-terminal nodes of a CS decision tree, takes the decision whether forwarding a sample to the left or the right child of a given node. However, as opposed to the case of standard decision trees, the decision is influenced not just by the single sample for which a decision has to be taken but also by the prediction hypotheses for all samples in the context. Indeed, we stress that the CS decision tree works given a set of samples, i.e. the context, and not simply a single sample. The split function is thus defined as $\varphi : \mathcal{X} \times 2^{\mathcal{X}\times\mathcal{Y}} \to \{0,1\}$, where the first argument is a sample in $\mathcal{X}$ and the second one is a subset of pairs in $\mathcal{X} \times \mathcal{Y}$ which represent the context samples with the respective prediction hypotheses.

As shown in Figure 5.2, the truncation of a CS decision tree at each node is obtained by exploiting the indexing imposed on the internal nodes of the tree. Given a CS decision tree $t \in \mathcal{T}$ and $m \in \mathbb{N}$, we denote by $t^{(<m)}$ a CS decision tree derived from t in which only the internal nodes having index $< m$ are kept and the internal nodes with index $\geq m$ having a parent with index $< m$, or being the root node, are converted into leaves. Finally, all nodes left-over are pruned away.

This procedure is inductively defined as follows:

$$\begin{aligned}(\textsc{Lf}\,(Q))^{(<m)} &= \textsc{Lf}\,(Q)\\ (\textsc{Nd}\,(n, Q, \varphi, t_l, t_r))^{(<m)} &= \begin{cases}\textsc{Lf}\,(Q) & \text{if } n \geq m\\ \textsc{Nd}\left(n, Q, \varphi, t_l^{(<m)}, t_r^{(<m)}\right) & \text{if } n < m\,.\end{cases}\end{aligned} \tag{5.1}$$

The basic idea here is that we can use $t^{(<m)}$ to obtain the prediction hypotheses for the context at a node indexed m.

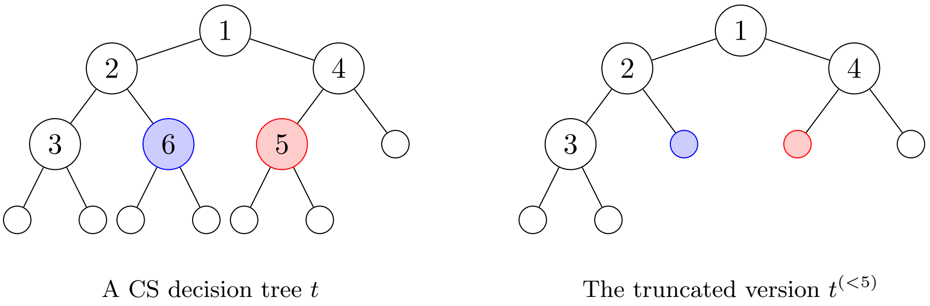


Figure 5.2: On the left, we find a CS decision tree t, where only the internal nodes are indexed. On the right, we see the truncated version $t^{(<5)}$ of t, which is obtained by converting to leaves all nodes having index ≥ 5 (we marked with colors the corresponding node transformations).

5.2.1 Inference.

The inference process, given a CS decision tree $t \in \mathcal{T}$, is equivalent to the one introduced for standard decision trees, with the only difference that a split function in a node indexed by n can use the truncated version of the same decision tree $t^{(<n)}$ to additionally exploit contextual information while taking decisions about where to route samples. Specifically, the posterior probability of $y \in \mathcal{Y}$ given a sample $x \in \mathcal{X}$ is inductively defined as:

$$P(y \,|\, x, t) = \begin{cases} Q(y) & \text{if } t = \textsc{Lf}\,(Q) \\ P(y \,|\, x, t_l) & \text{if } t = \textsc{Nd}\,(n, \cdot, \varphi, t_l, t_r) \text{ and } \varphi(x \,|\, t^{(<n)}) = 0 \\ P(y \,|\, x, t_r) & \text{if } t = \textsc{Nd}\,(n, \cdot, \varphi, t_l, t_r) \text{ and } \varphi(x \,|\, t^{(<n)}) = 1\,. \end{cases} \tag{5.2}$$

The same posterior probabilities with respect to a forest $\mathcal{F} \subseteq \mathcal{T}$ can be obtained as in (2.8).

5.2.2 Prioritized node training.

The training process for CS decision forests consists in training an ensemble of CS decision trees independently on random subsets of the training set $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$. Each CS decision tree is trained in an iterative way and, similar to the case of standard decision trees, a decision about whether to branch new nodes or produce a leaf is taken based on a subset of the training samples $\mathcal{Z} \subseteq \mathcal{D}$. However, in contrast to the standard setting, the learning process depends on the order in which nodes of the tree are grown because split functions depend on $t^{(<m)}$ which in turn is affected by the node ordering.

In other words, we impose here an explicit ordering on the recursive calls of function g performed in Equation (2.5). This ordering is determined by means of a priority queue, where the priority associated to each function call is determined according to a cost value. This cost can for instance be the depth at which a new node will be grown by the recursive call, in which case we enforce a breadth-first ordering, or the negative loss $-L(\mathcal{Z})$ defined as in (2.2), $\mathcal{Z}$ being the subset of the training data argument of the function call. This second option is particularly interesting because it forces the tree to split first the nodes where the training error measured in terms of the loss function is the highest. This indeed allows to reduce the uncertainty uniformly during the tree growth and thus have more reliable contextual information.

Whenever a new node is grown, it takes the time at which it was extracted from the priority queue as index. It is easy to see that the indexing deriving from this procedure never violates the property that children of a node have an index larger than the parent node. The split function selection is performed according to (2.6), the only difference being the type of split functions that are generated, which can exploit $t^{(<m)}$ to test contextual information.

We finally remark that the split functions might avoid explicit computations of posterior probabilities from the truncated trees by adopting a strategy like in [Montillo et al., 2011], in which the leaf where each training sample belongs to is tracked while growing the tree and a nearest neighbor approximation is done if samples not belonging to the training set are tested. A similar procedure can also be adopted during inference. In this case multiple samples are evaluated at the same time, by keeping track of their node position and by respecting a node evaluation order determined by the node's indices.

5.3 Application to Object Detection

In this section we employ the CS decision trees for the problem of object detection, following a solution setting similar to [Gall and Lempitsky, 2009]. Specifically, we adopt a patch-based abstraction of an image and the aim of the tree-based predictor is to jointly predict, for each patch, the foreground/background class it belongs to and a displacement vector pointing to the object's center. By collecting all the object position hypotheses from all foreground patches, we can setup a Hough space in which objects can be detected from the vote modes.

An image $I : \mathbb{Z}^2 \to \mathcal{F}$ is a function mapping pixels to elements of a feature space $\mathcal{F}$. The feature space here may include a variety of image cues, like color information, gradients, filter bank responses, *etc.* . We denote by $I(\mathbf{u}) \in \mathcal{F}$ the feature vector associated to pixel $\mathbf{u}$ and by $\mathcal{I}$ the set of images, and by $I(\mathbf{u})_k$ the kth element of the feature vector associated to $\mathbf{u}$. The input space $\mathcal{X}$ for our learning problem is a set of patches, each represented as a pair $(\mathbf{u}, I) \in \mathbb{Z}^2 \times \mathcal{I}$, pixel $\mathbf{u}$ being the center of the patch in image I.

The output space $\mathcal{Y}$ is a set of pairs $(c, \mathbf{d})$, where $c \in \{0, 1\}$ is a binary class label indicating the presence of an object and $\mathbf{d} \in \mathbb{Z}^2$ is the displacement of the object's center. Hence, if a training sample $(\mathbf{u}, I) \in \mathcal{X}$ has $(c, \mathbf{d}) \in \mathcal{Y}$ as the ground-truth prediction then we have in image I at location $\mathbf{u}$ either a background pixel $(c = 0)$ or a foreground pixel $(c = 1)$, i.e. belonging to an object and, if the second case holds, $\mathbf{u} + \mathbf{d}$ is the center of the object to which the pixel belongs. Note that $\mathcal{Y}$ encodes both the classification and regression part of the object detection task.

The loss function $\ell(c, \mathbf{d}|Q)$ that we employ for the computation of $L(\mathcal{Z}|Q)$ in (2.2) is given by

$$\ell(c, \mathbf{d} \,|\, Q) = \mathbb{E}_{(c',\mathbf{d}')\sim Q}\left[\mathbb{1}\left[c \neq c'\right] + \mathbb{1}\left[(c, c') = (1,1)\right]\left(1 - K_\sigma(\mathbf{d} - \mathbf{d}')\right)\right] \tag{5.3}$$

where $K_\sigma(\mathbf{x}) = \exp(-\|\mathbf{x}\|^2/\sigma^2)$. This quantity measures the expected loss that we incur by predicting $(c', \mathbf{d}')$ in place of $(c, \mathbf{d})$, where $(c', \mathbf{d}')$ is sampled according to Q. The term under expectation behaves as a $0/1$ loss for all combinations of class labels, excepting the case $c = c' = 1$ where also the correct prediction of the displacement vector is taken into account. Indeed, even if a pixel belonging to an object is correctly labelled, we incur in a high loss if the object's center position estimation is completely wrong. This is taken into account with the second term.

The density estimation function $\pi(\mathcal{Z})$, which generates the posterior distributions stored in the tree leaves, is different depending on whether we are at an internal node or at a leaf of the tree. In both cases it provides distributions that factorize in two marginal distributions, for the class labels and the displacement vector, respectively. Moreover, in both cases, the marginal over the class labels is a discrete distribution providing the probability of drawing a sample of a given class from the set $\mathcal{Z}$. The difference is with respect to the marginal over the displacement vector. We have a point-wise and uni-modal distribution at the internal nodes, while we keep track of multiple modes at the leaves.

Let $q \in \mathbb{P}(\{0, 1\})$ be the marginal distribution over the class labels defined as $q(c) = |\mathcal{Z}_c|/|\mathcal{Z}|$, where $\mathcal{Z}_0$ and $\mathcal{Z}_1$ are the set of background and foreground samples, respectively, in $\mathcal{Z}$. At the internal node level $\pi(\mathcal{Z})$ returns a probability distribution $Q_n \in \mathbb{P}(\mathcal{Y})$ defined as

$$Q_n\,(c, \mathbf{d}) = q(c)\delta(\mathbf{d} - \mathbf{d}^*)\,. \tag{5.4}$$

Here, $\mathbf{d}^*$ represents the single point-wise mode of the marginal distribution with respect to $\mathbf{d}$ (i.e. the second term), which is determined in a way to minimize the loss $L(\mathcal{Z}|Q_n)$ over the training samples. A local solution of the minimization problem can be found by iterating the following mean-shift [Comaniciu and Meer, 2002] procedure[1]

$$\mathbf{d}^* \leftarrow \sum_{(c,\mathbf{d})\in\mathcal{Z}_1} \mathbf{d}\, K_\sigma(\mathbf{d} - \mathbf{d}^*) \Big/ \sum_{(c,\mathbf{d})\in\mathcal{Z}_1} K_\sigma(\mathbf{d} - \mathbf{d}^*)\,. \tag{5.5}$$

An illustration for this mode finding procedure is given in Figure 5.3. The first plot shows the mean value (which would be used in the variance minimization objective of the standard Hough forest, cf. Equation (2.50)) of three randomly drawn Gaussians. Starting from this mean, the mode isolation procedure of Equation (5.5) selects one of the distributions as illustrated in the second plot. Doing so enables us to separate coherent

[1] In the experiments conducted, we never exceeded 10 iterations for finding a mode.

votes in a faster manner as opposed to the standard Hough forest in a small number of iterations (here, approximately seven).

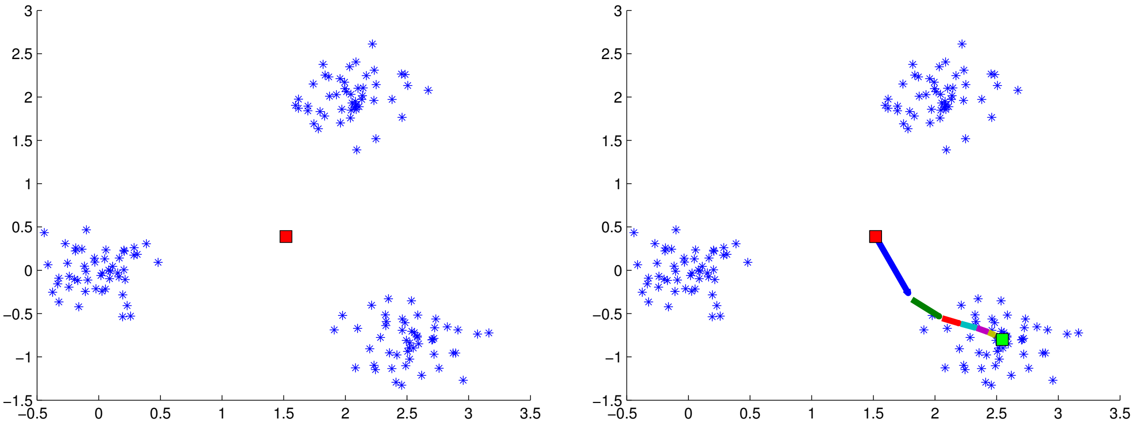


Figure 5.3: Illustration of mean-shift procedure, used for mode isolation in our context-sensitive decision trees. Left: Three Gaussians and mean (red) over all samples. Right: Starting from the mean, the mode isolation procedure picks one center after some iterations (green rectangle).

At the leaf level, instead, $\pi(\mathcal{Z})$ returns a probability distribution $Q_l \in \mathbb{P}(\mathcal{Y})$ defined as:

$$Q_l\left(c, \mathbf{d}\right) = q(c) \sum_{(c', \mathbf{d}') \in \mathcal{Z}_1} \delta(\mathbf{d} - \mathbf{d}')/|\mathcal{Z}_1|\,. \tag{5.6}$$

The second term, i.e. the marginal over $\mathbf{d}$, is uniform over the set of displacement vectors belonging to foreground samples reaching the leaf.

We define finally a novel type of split function, which performs a test by exploiting the contextual information. This test is particularly interesting because it allows to check whether two pixels are expected to belong to the same object instance. The new split function $\varphi^{(cs)}(\mathbf{u}, I|t, \mathbf{h}_1, \mathbf{h}_2, \tau)$ takes as input a sample $(\mathbf{u}, I) \in \mathcal{X}$ and it is parametrized by a CS decision tree $t \in \mathcal{T}$ that is used for generating the contextual information, by two relative displacement vectors $\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^2$ that identify the position of two pixels relative to $\mathbf{u}$ and by a threshold τ. The definition of our context-sensitive split functions $\varphi^{(cs)}$ is as follows:

$$\varphi^{(cs)}(\mathbf{u}, I|t, \mathbf{h}_1, \mathbf{h}_2, \tau) = \mathbb{1}\left[\mathbb{E}_{(c,\mathbf{d},c',\mathbf{d}') \sim P_1 \cdot P_2}\left[\mathbb{1}\left[(c, c') = (1,1)\right] K_\sigma(\mathbf{d} - \mathbf{d}')\right] < \tau\right] \tag{5.7}$$

where $P_j = P(\cdot|(\mathbf{u} + \mathbf{h}_j, I), t)$, with $j = 1, 2$, are the posterior probabilities obtained from tree t given samples at position $\mathbf{u} + \mathbf{h}_1$ and $\mathbf{u} + \mathbf{h}_2$ of image I, respectively. Please note that this test should not be confused with the regression split criterion in [Gall and Lempitsky, 2009], which tries to partition the training set in a way to group examples with similar voting direction and length. Besides the novel context-sensitive split function we employ also standard split functions performing tests on $\mathcal{X}$ as defined in Equation (2.20).

5.4 Experiments

To assess our proposed approach, we have conducted several experiments on the task of pedestrian detection. Detecting pedestrians is very challenging for Hough-voting based methods as they typically exhibit strong articulations of feet and arms, yielding to non-distinctive hypotheses in the Hough space. We evaluated our method on the TUD pedestrian data base [Andriluka et al., 2008] in two different ways: First, we show 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 (Section 5.4.1). With this experiment we show the improvement over state-of-the-art approaches when learning can be performed with simultaneous knowledge about context information. In a second variation (Section 5.4.2), we use the images of the Crossing scene (201 images) as a training set. Most images of this scene contain more than four persons with strong overlap and mutual occlusions. However, instead of using the original annotation which covers only pedestrians with at least 50% overlap (1008 bounding boxes), we use the more accurate, pixel-wise ground truth annotations of [Riemenschneider et al., 2012] for the entire scene that includes all persons and consists of 1215 bounding boxes. Please note that this ground truth is even more detailed than the one presented in [Barinova et al., 2010] with 1018 bounding boxes. The purpose of the second experiment is to show that our context-sensitive forest can exploit the availability of multiple training instances significantly better than state-of-the-art.

The most related work and therefore also the baseline in our experiments is the Hough Forest [Gall and Lempitsky, 2009]. To guarantee a fair comparison, we use the same training parameters for [Gall and Lempitsky, 2009] and our context sensitive forest: We trained 20 trees and the training data (including horizontally flipped images) was sampled homogeneously per category per image. The patch size was fixed to 30×30 and we performed 1600 node tests for finding the best split function parameters per node. The trees were stopped growing when < 7 samples were available. As image features, we used the the first 16 feature channels provided in the publicly available Hough Forest code of [Gall and Lempitsky, 2009] which include CIELab raw channel intensities, first and second order derivatives as well as HOG-like features, computed on the L-Channel. In order to obtain the object detection hypotheses from the Hough space, we use the same Non-maximum suppression (NMS) technique in all our experiments as suggested in [Gall and Lempitsky, 2009]: After smoothing is performed in the Hough scale space frustum, maxima are extracted until the quality of the best hypothesis cannot exceed a pre-specified threshold. Each detected maximum is removed and suppresses adjacent regions with a certain window size in the scale space. 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 $\geq 50\%$, using the publicly available Matlab implementation of Dollár[2]. The additional parameter of (5.3) was set to $\sigma = 7$. 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.

5.4.1 Evaluation using standard protocol training set

The standard training set contains 400 images where each image comes with a single pedestrian annotation. 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 $2 \cdot 400 \cdot 2 \cdot 125 = 200k$ training samples per tree. For additional comparisons, we provide the results presented in the recent work on joint object detection and segmentation of [Riemenschneider et al., 2012], from which we also provide evaluation results of the Implicit Shape Model (ISM) [Leibe et al., 2008]. However, please note that the results of [Riemenschneider et al., 2012] are based on a different baseline implementation. Moreover, we show the results of [Barinova et al., 2010] when using the provided code and configuration files from the first authors homepage. Unfortunately, we could not reproduce their results of the original work.

First, we discuss the results obtained on the Campus scene. This data set consists of 71 images showing walking pedestrians at severe scale differences and partial occlusions. The ground truth we use has been released with [Barinova et al., 2010] and contains a total number of 314 pedestrians. Figure 5.4, first row, plot 1 shows the precision-recall curves when using 3 scales (factors 0.3, 0.4, 0.55) for our baseline [Gall and Lempitsky, 2009] (blue), results from re-evaluating [Barinova et al., 2010] (cyan, 5 scales), [Riemenschneider et al., 2012] (green) and our Context-Sensitive Forest without and with using the priority queue based tree construction (red/magenta). In case of not using the priority queue, we trained the trees according to a breadth-first way. We obtain a performance boost of $\approx 6\%$ in recall at a precision of 90% when using both, context information and the priority based construction of our forest. The second plot in the first row of Figure 5.4 shows the results when the same forests are tested on the Crossing scene, using the more detailed ground truth annotations. The data set shows walking pedestrians (Figure 5.5, top row images) with a smaller variation in scale compared to the Campus scene but with strong mutual occlusions and overlaps. The improvement with respect to the baseline is lower ($\approx 2\%$ gain at a precision of 90%) and we find similar developments of the curves. However, this comes somewhat expectedly as the training data does not properly reflect the occlusions we would like to have modelled for this data set. With our next experiment we demonstrate the effect when learning from training data with occluded training instances.

[2] `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`

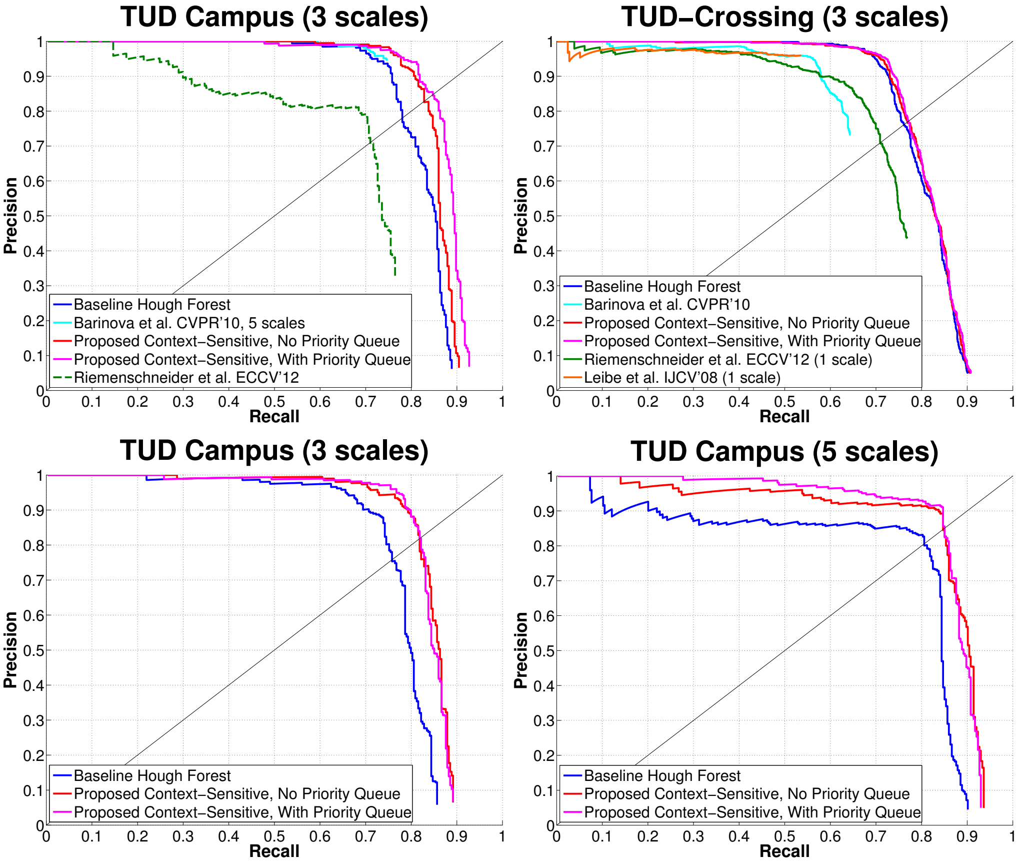


Figure 5.4: Precision-Recall Curves for Detections, Top row: Standard training (400 images), evaluation on Campus and Crossing (3 scales). Bottom row: Training on Crossing annotations of [Riemenschneider et al., 2012], evaluation on Campus, 3 and 5 scales.

5.4.2 Evaluation with Crossing scene as training set

In our next experiment we trained the forests (same parameters) on the annotations of [Riemenschneider et al., 2012] for the Crossing scene, reducing the training set to only 201 images. Qualitative detection results are shown in the bottom row images of Figure 5.5. From the first precision-recall curve in the second row of Figure 5.4 we can see, that the margin between the baseline and our proposed method could be improved (gain of $\approx 9\%$ recall at precision 90%) when evaluating on the same 3 scales. With evaluation on 5 scales (factors 0.34, 0.42, 0.51, 0.65, 0.76) we found a strong increase in the recall, however, at the cost of loosing $2 - 3\%$ of precision below a recall of 60%, as illustrated in the second plot of row 2 in Figure 5.4. While our method is able to maintain a precision above 90% up to a recall of $\approx 83\%$, the baseline implementation drops already at a recall of $\approx 20\%$. From this experiment we conclude that the integration of contextual information into the learning process leads to improved detection results, especially when the training data contains multiple instances of the desired object category.

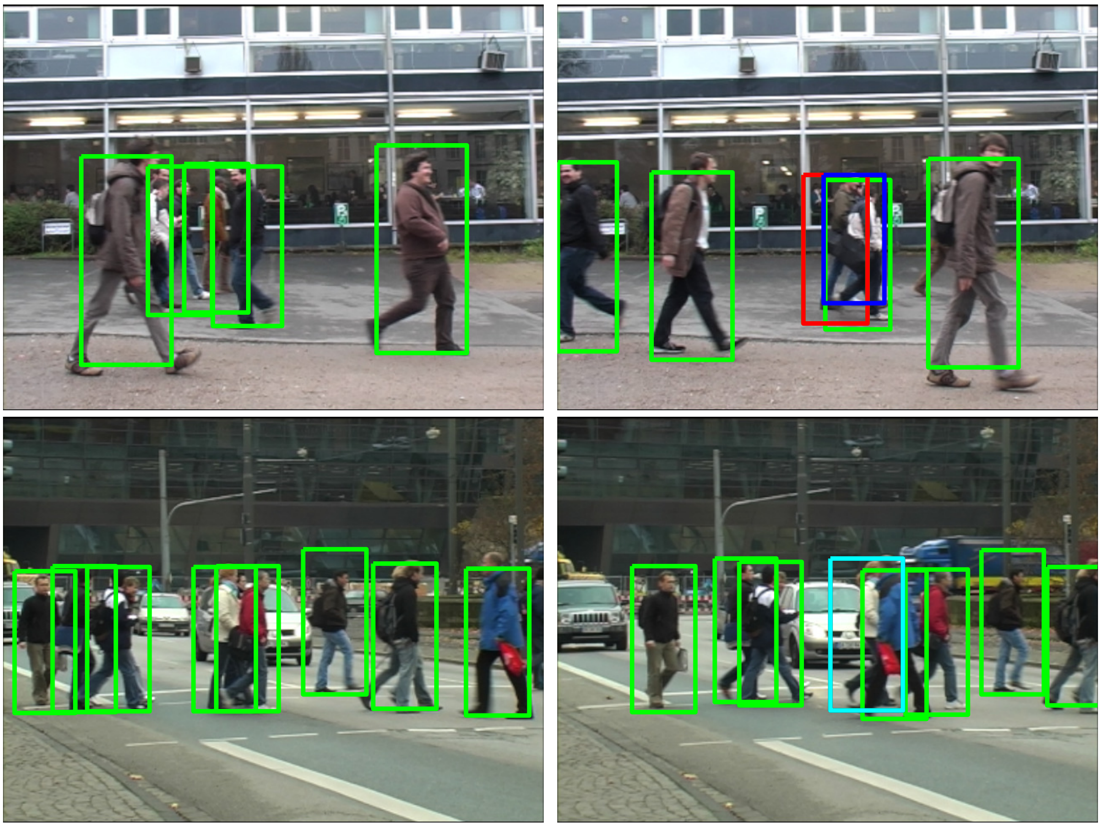


Figure 5.5: Qualitative examples for Campus (Top row) and Crossing (Bottom Row) scenes. (green) correctly found by our method (blue) ground truth (red) wrong association (cyan) missed detection.

5.5 Conclusion

In this chapter we have presented Context-Sensitive Decision Forests with application to the object detection problem. Our new forest has the ability to access intermediate prediction (classification and regression) information about all samples of the training set and can therefore learn from contextual information in the growing process. This is in contrast to existing random forest methods used for object detection which typically treat training samples in an independent manner. Moreover, we have introduced a novel splitting criterion together with a mode isolation technique, which allows us to (a) perform a priority-driven way of tree growing and (b) install novel context-based test functions to check for mutual object centroid agreements. In our experimental results on pedestrian detection we demonstrated superior performance with respect to state-of-the-art methods and additionally found that our new algorithm can better exploit training data containing multiple training objects.

Part II

Semantic Segmentation with Random Decision Trees

Chapter 6

Structured Labels for Semantic Segmentation and Object Detection

Contents

In this chapter, we propose a simple and effective way to integrate contextual information in random forests, which is typically reflected in the structured output space of semantic image labelling problems. By structural information we refer to the inherently available, topological distribution of object classes in a given image. Different object class labels will not be randomly distributed over an image but usually form coherently labelled regions. 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 performances on the task of object detection.

6.1 Introduction

Given the importance of context as already discussed in Chapter 5 of this Thesis, the aim of the current chapter is to enable the use of contextual information within the random forest framework especially for the problem of semantic image labelling. However, before providing more details about the contribution of this chapter in Section 6.1.2, we give an overview on more recent works that are using context for semantic segmentation in computer vision.

6.1.1 Related Work

In the last years, the mainstream of object categorization frameworks have adopted *graphical models* to model context due to their flexibility in encoding structure of local dependencies and incorporating contextual features with appearance-based detectors. A few works are based on *directed graphical models* [Russell et al., 2007, Shinghal et al., 2003, Torralba, 2003], while the majority exploits *undirected graphical models*, such as Markov Random Fields (MRF) [Carbonetto et al., 2004] or Conditional Random Fields (CRF) [He et al., 2004b, Kumar and Hebert, 2005, Rabinovich et al., 2007, Shotton et al., 2007, Torralba et al., 2004, Verbeek and Triggs, 2008, Ladicky et al., 2010a, Ladicky et al., 2010b, Gonfaus et al., 2010]. 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 [Ladicky et al., 2010a, Ladicky et al., 2010b, Gonfaus et al., 2010] typically incorporate complementary features at different levels. Low-level features are mostly calculated on a per-pixel basis and incorporate local color 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 [Biederman, 1972], 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) [Nowozin et al., 2011], where a conditional random field instance is generated for each test image under the guidance of trained decision trees, holding the CRF's parameters. Further improvements of this work can be found in [Jancsary et al., 2012]. Another example of combination of random forests with random fields can be found in [Payet and Todorovic, 2012].

Probabilistic graphical models, and in particular random fields, fall within the broader class of structured prediction models. Another approach based on structured learning theory, which allows to learn contextual relations is proposed in our work in [Rota Bulò et al., 2012]. We based our work on 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 neighboring pixels. Another work is [Zhu et al., 2008a], where an approach to semantic segmentation is proposed trying 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. Finally, we mention the work in [Yang et al., 2012], which introduces the concept of a label descriptor guiding the alignment of label patches, which in turn are forced to mutually agree and correlate with the local feature descriptors within an image. We refer to [Nowozin and Lampert, 2011] for a comprehensive tutorial on structured learning and prediction in computer vision.

A different way to effectively learn a contextual model has been presented in [Tu, 2008] with an approach named *Auto-context*. In this work, the author trained a sequence of classifiers in a boosting-fashion using both appearance-based features and contextual information obtained by the classifiers itself. The learning phase is however computationally very demanding. Other approaches like boosted random fields [Torralba et al., 2005] or SpatialBoost [Avidan, 2006] share both the disadvantage of significant computational complexity when considering contextual beliefs as weak learners. Similar to [Tu, 2008], the Texton forest of [Shotton et al., 2008b] introduced a model using context information, but for the first time in the random forest framework.

6.1.2 Contribution

In this chapter we show how to exploit structural information about the labelling output space within the random forest framework. Despite their popularity, very few works have tried to exploit contextual and structural information in random forests in order to improve their performance. In [Montillo et al., 2011], the authors propose *entangled decision forests* for semantic image labelling, which basically integrate an auto-context model [Tu, 2008] into the random forest framework. In the previous chapter of this Thesis we have shown how a related approach called *context-sensitive forest* can also improve the performance for object detection tasks.

In the setting of this chapter, we depart from the standard classification setting, in which a single (atomic) label is associated to each training sample, and we take structured label information from each pixel's neighborhood into account. As an example, consider the problem of classifying pixels into semantic categories as shown in Figure 6.1. On the bottom left, we can see some training samples that can be used to train a standard random forest. Each sample consists of a patch of the image centered 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 centered on each sample. Differently from previous approaches, we are moving from an unstructured output space to a structured output space. We describe this transition in Section 6.2.1. From a methodological point of view, we show how random forests can be adapted in order to take effectively and efficiently advantage of this additional information. In Section 6.2.2, we provide a new test function selection criterion which allows to exploit the information of structured label patches along the tree growing procedure, and in Section 6.2.3 we show how to obtain a structured prediction for each pixel, from a trained forest. In Sections 6.2.4 and 6.2.5 we propose also some mechanisms to integrate the proposed structured predictions over pixels of an image into a coherent labelling.

In order to address object detection tasks with the proposed approach, we perform a generalized Hough transform driven by our forest. The approach follows [Gall et al., 2011] with the important difference that contextual information about the labelling is exploited during training. Details can be found in Section 6.3. From the experimental perspective, we show that by including contextual information at the classification level within the random forest, the results improve for applications like semantic image labelling and object detection. Indeed, the structured output space allows to counteract the assignment of (semantically) meaningless label configurations, as experienced when using standard random forests. This is due to the fact that our forest can learn the oriented, spatial label distribution characterizing the ground-truth labellings and avoid implausible label transitions during the inference stage. Section 6.4 is devoted to the experimental evaluation of the proposed approach. In Sections 6.4.1 and 6.4.2 we perform experiments on several semantic image labelling datasets like CamVid and MSRCv2 and in 6.4.3 we show how to address the problem of inpainting by reconstructing occluded handwritten Chinese

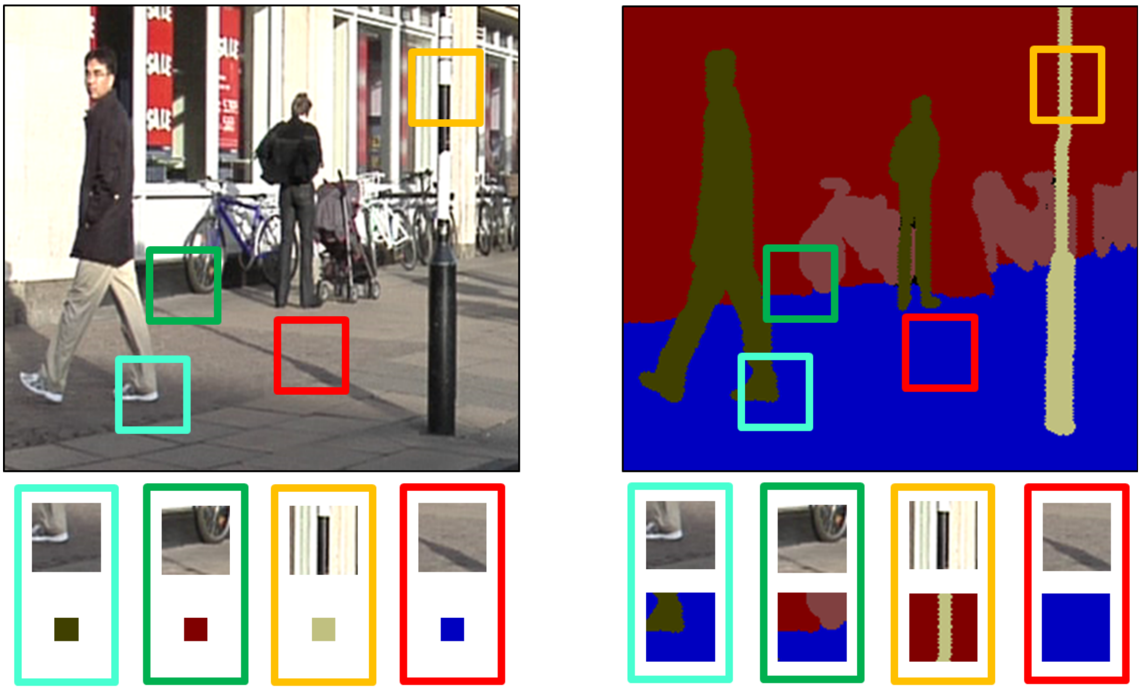


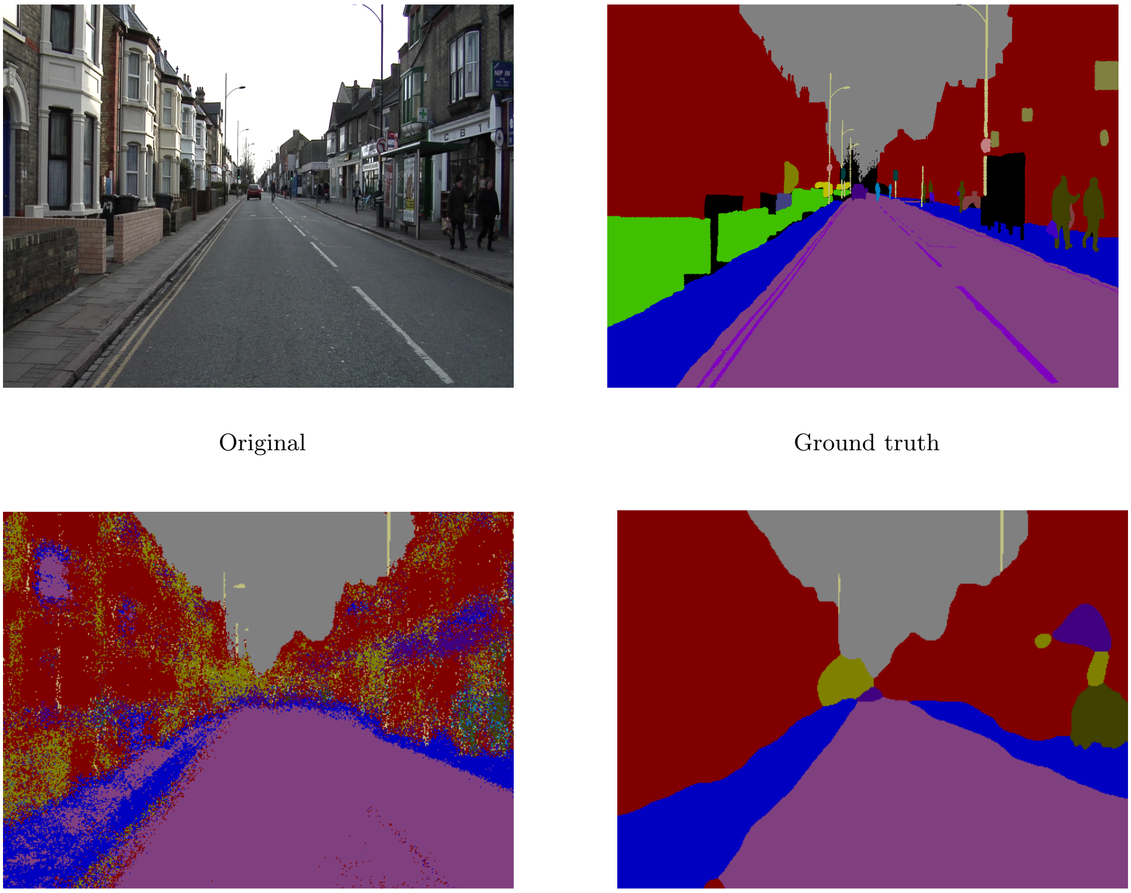
Figure 6.1: Comparison of class labels used in standard random forest and presented structured class label random forest. Top: Example Training image with corresponding label image and color-coded rectangles indicating training sample locations. Bottom: Associated, atomic class labels used in standard random forests and proposed structured class labels.

characters. Sections 6.4.4 - 6.4.6 are devoted to evaluating the performance of our forest on the task of pedestrian detection on different TUD datasets.

6.2 Structured Learning in Random Forests

In traditional classification approaches like the one presented in Chapter 2.2 of this Thesis, 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 inherently has a correlated structure, rendering 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 forests introduced in the previous sections. Consequently, when applying standard random forest classifiers for semantic image labelling, the obtained results are quite noisy (e.g. see Figure 6.2). 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



Original

Ground truth

Conventional Classification Forest

Our method

Figure 6.2: Examples of object class segmentations using unary classifiers. Best viewed in color.

forest classifiers by rendering them aware of the local topological structure of the output label space, as indicated in Figure 6.1. Towards this end, we depart from the traditional classification paradigm and address the problem from a structured learning perspective [Tsochantaridis et al., 2004] within the random forest framework.

6.2.1 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 $p : \mathbb{Z}^2 \to \mathcal{Y} \cup \{\perp\}$ providing a class label in $\mathcal{Y}$, or $\perp$ in case no label is assigned, to every pixel. With $p(\mathbf{d})$ we denote the entry of the label patch $p \in \mathcal{P}$ located at $\mathbf{d} \in \mathbb{Z}^2$. 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 general notation in Chapter 2.3 on page 11) 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 $\mathbf{x} = (\mathbf{u}, I)$ has an associated label patch p which holds the labels of all pixels of image I within the neighborhood of $\mathbf{u}$ determined implicitly by the patch. Figure 6.3 shows an example of a square feature patch x and an associated square label patch centered 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.

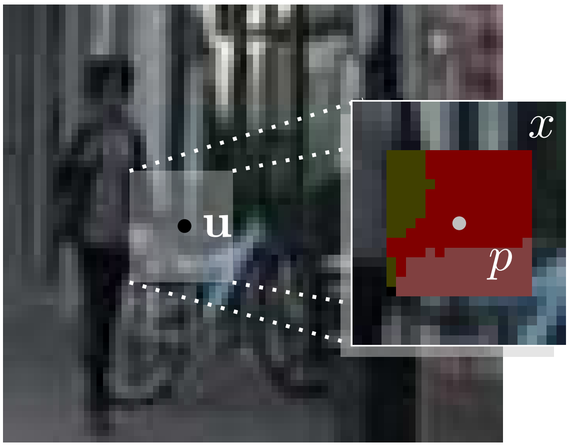


Figure 6.3: Training data example, as used in our proposed structured learning random forest. While standard random forests associate only the center label at (u, v) to a patch $\mathbf{x}$, we incorporate the local label neighborhood $\mathbf{p}$ and learn valid labelling transitions among adjacent object categories (here: person, building and bicycle).

In the next subsection 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 $\mathcal{P}_t \subseteq \mathcal{P}$ be the set of label patches present in the training data used to grow the leaf (see Figure 6.16). The class label π parameterising 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 joint distribution of the label patches in $\mathcal{P}_t$. We compute the joint probability by making a position-dependent pixel independence assumption as

$$\mathsf{P}[p|\mathcal{P}_t] = \prod_{\mathbf{d}} \mathsf{P}^{(\mathbf{d})}[p(\mathbf{d})|\mathcal{P}_t]\,, \tag{6.1}$$

where $\mathsf{P}^{(\mathbf{d})}[\cdot|\mathcal{P}_t]$ represents the marginal class distribution over all the label patches in $\mathcal{P}_t$ of labels located in position $\mathbf{d}$ and the product is taken over all positions $\mathbf{d} \in \mathbb{Z}^2$ such that $p(\mathbf{d}) \neq \perp$. In such a way, we keep the complexity of this step low but simultaneously consider the topological label statistics at absolute positions in the label patches. Finally, the label patch π selected for leaf t is the one in $\mathcal{P}_t$ maximizing the joint probability, i.e. the candidate in $\mathcal{P}_t$ that is closest to the location-dependent joint probability as defined above:

$$\pi \in \arg\max_{p \in \mathcal{P}_t} \mathtt{P}[p|\mathcal{P}_t] \,. \tag{6.2}$$

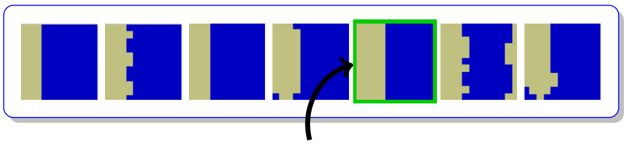


Figure 6.4: Example of label patches reaching a leaf during training. Based on the joint probability distribution of labels in the leaf a label patch π is selected.

6.2.2 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 center 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 new label space.

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 $\mathbf{d}$, which is generated once per node, instead of considering as usually done the label of the central pixel.

In Figure 6.5 we provide an illustration when considering the case $k = 2$, i.e. we fix the center 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 our proposed structured class labels allows to additionally consider contextual constraints from the enhanced label space.

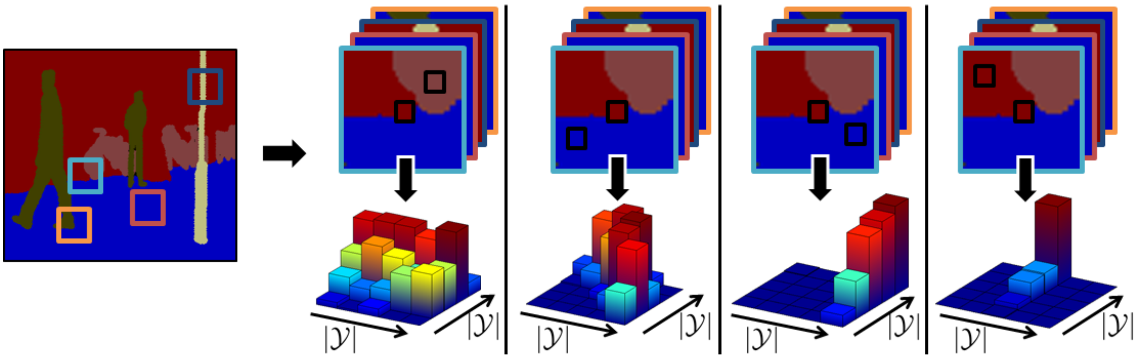


Figure 6.5: Left: Ground truth label image with training sample locations. Right: Schematic illustrations of resulting $k = 2$ joint label distributions evaluated on split candidate sets for different label positions.

6.2.3 Structured Label Predictions

The structured predictions gathered from the trees of a forest have to be combined into a single label patch prediction. To this end, we follow a procedure which is similar to the one adopted in order to select the label patch π in a leaf (see Section 6.2.1).

Consider a trained forest $\mathcal{F}$, a test patch $x = (\mathbf{u}, I)$ and let $\mathcal{P}_\mathcal{F}$ be the set of predictions for x gathered from each tree $t \in \mathcal{F}$:

$$\mathcal{P}_\mathcal{F} = \{h(x|t) \in \mathcal{P} \,:\, t \in \mathcal{F}\}\,. \tag{6.3}$$

Similarly to (6.2), the label patch prediction of the forest $\mathcal{F}$ for feature patch x is given by the one maximizing the patch label joint probability estimated from $\mathcal{P}_\mathcal{F}$, i.e.

$$p^* \in \arg\max_{p \in \mathcal{P}_\mathcal{F}} \mathtt{P}[p \,|\, \mathcal{P}_\mathcal{F}]\,, \tag{6.4}$$

where $\mathtt{P}[p|\mathcal{P}_\mathcal{F}]$ is defined as in (6.1).

6.2.4 Simple Fusion of Structured Predictions

As opposed to standard classification algorithms which, given a test image I, directly assign an object class label to a each pixel, our classifiers cast a prediction for each pixel, involving also the neighboring ones. Indeed, if $p \in \mathcal{P}$ is the patch label predicted for pixel

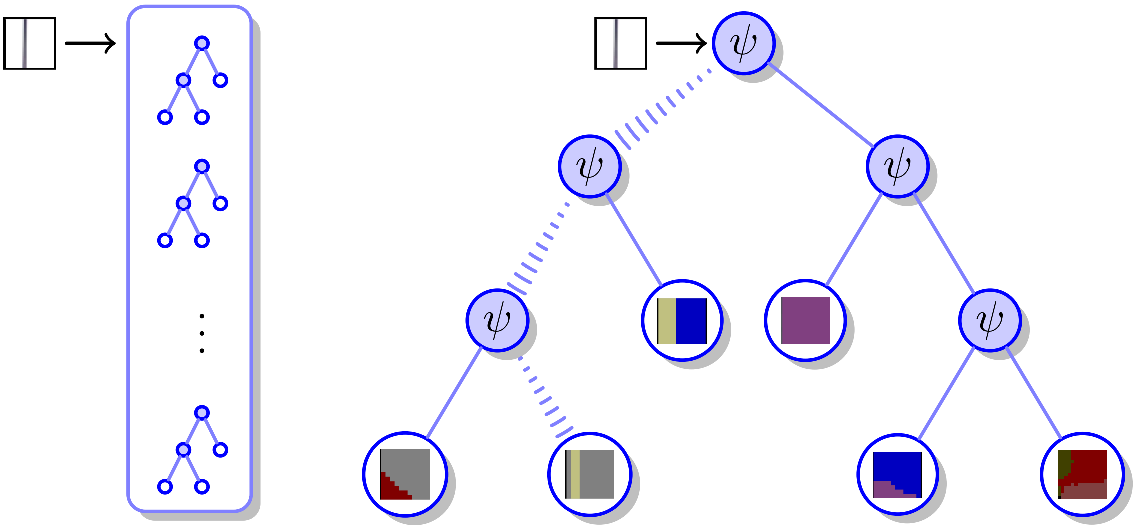


Figure 6.6: Prediction of the structured label of a feature patch in a random forest. The feature patch is routed through each tree in the forest according to the test functions ψ in the tree nodes until a leaf is reached, holding the learned label transitions between color-coded classes. The structured label in the leaf is then assigned to the feature patch. Best viewed in color.

$\mathbf{u}$ in a test image then a pixel $\mathbf{v}$ in a neighborhood 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 neighboring pixels, which have to be integrated into a single class prediction. This process is illustrated in Figure 6.7. As we can see, assuming 3×3 square-shaped label patches, each test pixel receives 9 class predictions from the neighborhood, 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 ℓ 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 $\mathbf{u}$.

6.2.5 Optimizing the Label Patch Selection

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 (6.4) for each pixel. This allows to better exploit the label patch diversity in the set of predictions $\mathcal{P}_\mathcal{F}$ obtained from (6.3).

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

$$\phi^{(\mathbf{v})}(p,\ell) = \sum_{\mathbf{u}\in\mathrm{dom}(I)} \mathbb{1}\left[p(\mathbf{u}-\mathbf{v}) = \ell(\mathbf{u})\right]. \tag{6.5}$$

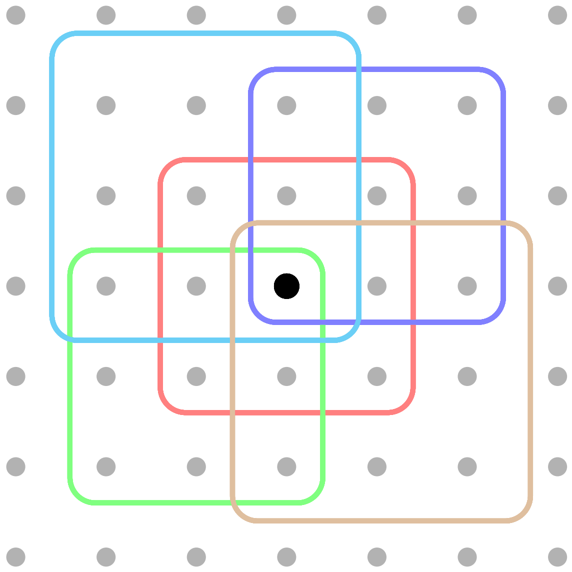


Figure 6.7: Simple fusion of structured predictions using 3 × 3 label patches. Each pixel collects class hypotheses from the structured labels predicted for itself and neighboring pixels, which have to be fused into a single class prediction. For clarity reasons, only 5/9 label patches are drawn. Best viewed in color.

Furthermore, let $z \in \mathcal{Z}_I$ be an assignment of label patches to pixels in I, $z_{\mathbf{v}} \in \mathcal{P}_F$ being a label patch for pixel $\mathbf{v}$ taken from (6.3), where $\mathcal{Z}_I$ denotes the set of all such assignments for image I. Then, for a particular configuration $z \in \mathcal{Z}_I$ 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

$$\Phi(z, \ell) = \sum_{\mathbf{v} \in \mathrm{dom}(I)} \phi^{(\mathbf{v})}(z_{\mathbf{v}}, \ell)\,. \tag{6.6}$$

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 $(z^*, \ell^*) \in \mathcal{Z}_I \times \mathcal{L}$, where

$$(z^*, \ell^*) \in \operatorname*{arg\,max}_{(z,\ell) \in \mathcal{Z}_I \times \mathcal{L}} \Phi(z, \ell)\,. \tag{6.7}$$

The optimization problem in (6.7) underlying our image labelling approach is in general non-trivial to solve. The algorithm we propose is a heuristic, which is simple and effective as shown in the experiments conducted. 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 $z \in \mathcal{Z}_{\mathcal{I}}$ given a labelling for image $\mathcal{I}$.

Let $\ell^{(t)}$ be the labelling of the image at a given time $t \geq 0$. The configuration of label patches $z^{(t+1)}$ at time $t+1$ can be obtained according to the following updating scheme:

$$z_{\mathbf{v}}^{(t+1)} \in \arg\max_{p \in \mathcal{P}_{\mathcal{F}}} \phi^{(\mathbf{v})}\left(p, \ell^{(t)}\right) \tag{6.8}$$

where $\mathcal{P}_{\mathcal{F}}$ is the set of label patches gathered from the forest $\mathcal{F}$ for pixel $\mathbf{v}$ according to (6.3). This updating equation selects for each pixel $\mathbf{v} \in \mathrm{dom}(I)$ a label patch in the set $\mathcal{P}_{\mathcal{F}}$ of available patches maximizing the agreement with the labelling $\ell^{(t)}$. On the other hand, given the configuration of patches $z^{(t+1)} \in \mathcal{Z}_{\mathcal{I}}$ for image $\mathcal{I}$ at time $t+1$, we compute the new labelling $\ell^{(t+1)} \in \mathcal{L}$ by taking a majority vote over all label patches as follows:

$$\ell^{(t+1)}(\mathbf{u}) \in \arg\max_{y \in \mathcal{Y}} \left\{ \sum_{\mathbf{v} \in \mathrm{dom}(I)} \mathbb{1}\left[z_{\mathbf{v}}^{(t+1)}(\mathbf{u}-\mathbf{v}) = y\right] \right\}. \tag{6.9}$$

The iterative process is started from an initial labelling $\ell^{(0)} \in \mathcal{L}$ and, by repeatedly applying rules (6.8) and (6.9), it will eventually converge towards a local solution of (6.7). Theorem 3, indeed, guarantees that the iterative scheme never decreases the value of the objective function Φ.

Proposition 3. *Let $I \in \mathcal{I}$ be an image and let $\ell^{(0)} \in \mathcal{L}$ be an initial labelling for I, and let $z^{(0)} \in \mathcal{Z}_{\mathcal{I}}$ be an initial configuration of label patches. Then for any $t \geq 0$ we have*

$$\Phi\left(z^{(t+1)}, \ell^{(t)}\right) \geq \Phi\left(z^{(t)}, \ell^{(t)}\right) \tag{6.10}$$

and

$$\Phi\left(z^{(t+1)}, \ell^{(t+1)}\right) \geq \Phi\left(z^{(t+1)}, \ell^{(t)}\right) \tag{6.11}$$

where $z^{(t+1)}$ and $\ell^{(t+1)}$ are computed according to (6.8) *and* (6.9)*, respectively.*

Proof. By (6.8) we have for all $t \geq 0$ and $\mathbf{v} \in \mathrm{dom}(I)$:

$$\phi^{(\mathbf{v})}\left(z_{\mathbf{v}}^{(t+1)}, \ell^{(t)}\right) \geq \phi^{(\mathbf{v})}\left(z_{\mathbf{v}}^{(t)}, \ell^{(t)}\right).$$

By summing up each side of this inequality for all pixels $\mathbf{v} \in \mathrm{dom}(I)$ we obtain (6.10).

As for the second inequality, note that by (6.9) and (6.5) we have

$$\sum_{\mathbf{v} \in \mathrm{dom}(I)} \mathbb{1}\left[z_{\mathbf{v}}^{(t+1)}(\mathbf{u}-\mathbf{v}) = \ell^{(t+1)}(\mathbf{u})\right] \geq \sum_{\mathbf{v} \in \mathrm{dom}(I)} \mathbb{1}\left[z_{\mathbf{v}}^{(t+1)}(\mathbf{u}-\mathbf{v}) = \ell^{(t)}(\mathbf{u})\right]$$

for all $\mathbf{u} \in \mathrm{dom}(I)$. This together with a trivial re-ordering of the summations yields

$$\begin{aligned}\Phi\left(z^{(t+1)}, \ell^{(t+1)}\right) &= \sum_{\mathbf{u}\in\mathrm{dom}(I)} \sum_{\mathbf{v}\in\mathrm{dom}(I)} \mathbb{1}\left[z_{\mathbf{v}}^{(t+1)}(\mathbf{u}-\mathbf{v}) = \ell^{(t+1)}(\mathbf{u})\right] \\ &\geq \sum_{\mathbf{u}\in\mathrm{dom}(I)} \sum_{\mathbf{v}\in\mathrm{dom}(I)} \mathbb{1}\left[z_{\mathbf{v}}^{(t+1)}(\mathbf{u}-\mathbf{v}) = \ell^{(t)}(\mathbf{u})\right] = \Phi\left(z^{(t+1)}, \ell^{(t)}\right)\end{aligned}$$

from which the result derives. □

As for the computational complexity of the solver, let N be the number of pixels, K the average number of label patches per pixel, M the number of non-void elements of a label patch, k the number of labels, and γ the number of iterations. An update step for the pixel configuration z has complexity $O(K \cdot M \cdot N)$, while an update step for the labelling ℓ has complexity $O\left((k + M) \cdot N\right)$. The overall complexity is thus given by $O\left(\gamma \cdot (k + M + KM) \cdot N\right)$. Note that in our experiments we stopped the iterative process if either a fixed point or a maximum number $\gamma = 75$ of iterations was reached.

A simple way to speedup the entire optimization process is to keep track of areas in the image where different label patches are selected and/or which positions are still affected in the fusion step. When neither of the alternatingly executed processes induces any changes, this particular region must not be updated any more and can therefore be ignored throughout the rest of the optimization process.

6.3 Enhancing Object Detection with Structured Class-Labels

As described in the first part of this Thesis, the random forest framework has been successfully applied for object detection, i.e. the problem of localizing multiple instances of a specific object class within an image. When using the Hough forest as in the previous part, we restricted the categorical label space of the learner to single class labels for each training sample. Here, we present a modification to cope with label patches instead of atomic class labels. As we show in the experimental section, by exploiting the additional information delivered by the label patches, we achieve a considerable improvement on the generalization capabilities of the forest, achieving state-of-the-art on standard datasets.

We refrain from repeating the algorithmic underpinnings of the Hough forest here and instead refer to Chapter 2.6.3 of this Thesis. Interestingly, we can leave the inference process completely unchanged (as we do not care about the structured, categorical predictions), and instead simply exchange the objective function considering the class-label uncertainty (see Equation (2.49)) with the one described in Section 6.2.2 of this chapter. In such a way, the more sophisticated examination of structured label patches during training is used to improve the detection results, as we demonstrate with our experiments on pedestrian detection in Section 6.4.4.

6.4 Experiments

In this section we evaluate our proposed structured learning random forest algorithm for the task of semantic segmentation on the CamVid [Brostow et al., 2008], MSRCv2 [Shotton et al., 2006] and KAIST Hanja2[1] databases and 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 label patch size of 1×1 and a fixed label center position. Where available, we list results of state-of-the-art methods [Shotton et al., 2006, Brostow et al., 2008, Kluckner et al., 2009] that are also using random forests (but not the same image features), in order to show that our baseline random forest implementation achieves state-of-the-art performance. Additionally, we compare to the results obtained when minimizing the energy term of a pairwise, conditional random field (CRF) model with graph cuts, using the publicly available GCO [Boykov et al., 2001] implementation[2]. To this end, we provide the class label statistics of the baseline random forest as unary or data terms and use the standard, contrast-sensitive Potts model as suggested in [Boykov and Jolly, 2001] for the pairwise or smoothness term.

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' considers the structured label patches but only takes one random label position, i.e. a single label distribution into account for training. 'k-Full' considers structured label patches and k-label joint distributions in the split functions (see Section 6.2.2 for more details). 'Simple Fusion' and 'Optimized Selection' refer to the fusion methods of the structured output predictions as described in Sections 6.2.4 and 6.2.5, respectively.

We used the same low-level image features for training both, our baseline and our novel structured learning random forests, since our primary intention is to show the improvement when the extended label space is taken into account: 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 gray scale intensity image, similar to [Porikli et al., 2006, Kluckner et al., 2009]. Please note that our presented method is independent with respect to the number and types of used feature cues. 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.

[1] http://ai.kaist.ac.kr/Resource/dbase/Hanja/HanjaDB2.htm

[2] http://vision.csd.uwo.ca/code/

We list the scores of our experiments according to the same evaluation criteria as used in [Shotton et al., 2006, Brostow et al., 2008, Kluckner et al., 2009] and additionally include the more strict average intersection vs. union score as e.g. used in the PASCAL VOC challenges [Everingham et al., 2010]. In particular, '*Global*' refers to the percentage of all pixels that were correctly classified, '*Avg(Class)*'[3] expresses the average recall over all classes and '*Avg(Pascal)*'[4] denotes the average intersection vs. union score (sometimes also referred to as Jaccard measure).

6.4.1 CamVid Database Experiments

The Cambridge-driving Labeled Video Database (CamVid) [Brostow et al., 2008] 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 [Brostow et al., 2008, Sturgess et al., 2009].

We resized the training 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. 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.

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%): Road, Building, Sky, Tree, Sidewalk, Car, Column_Pole, Sign-Symbol, Fence, Pedestrian and Bicyclist. In Table 6.1 we list our results using a label patch size of 13 × 13, clearly indicating the performance boost when using our proposed structured learning method over the standard random forest. We can achieve comparable results to the CRF implementation with the Simple Fusion approach and significantly increase the scores using the Optimized Selection. We explain this by the fact that our method is restricted to pick from a candidate set of semantically plausible label patches provided by the trees, rather than allowing to propagate arbitray label configurations in the associated graphical model.

In Figure 6.8 we show the influence of the label patch size during training and classification using the configuration '2-Full + Simple Fusion'. It is clearly shown that even a small neighborhood ($\geq 5 \times 5$) leads to a significant boost in the classification stage.

[3] $\frac{\text{True Positives}}{\text{True Positives + False Negatives}}$

[4] $\frac{\text{True Positives}}{\text{True Positives + False Negatives + False Positives}}$

Method	*Global*	*Avg(Class)*	*Avg(Pascal)*
RF using Motion and Structure cues [Brostow et al., 2008]	61.8	43.6	-
RF using Motion and Structure cues [Brostow et al., 2008] + Appearance features	69.1	53.0	-
Local label descriptors [Yang et al., 2012]	73.7	36.3	29.6
Our Baseline RF	69.9	42.2	30.6
Our Baseline RF + CRF	74.5	45.4	33.8
Our method (Structure + Simple Fusion)	74.8	45.0	34.1
Our method (2-Full + Simple Fusion)	76.8	46.1	35.4
Our method (2-Full + Optimized Selection)	79.2	46.0	36.2
Our method (2-Full + Optimized Selection) + Correlation coefficients	83.8	53.2	43.5
Our method (3-Full + Optimized Selection) + Correlation coefficients	82.0	52.1	41.6

Table 6.1: Classification results on CamVid database for label patch size 13 × 13 and comparisons to related works.

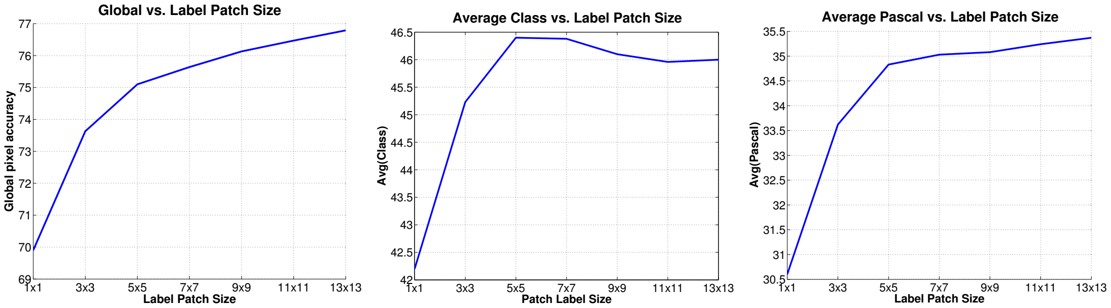


Figure 6.8: Classification results in terms of global, per-class average and average intersection vs. union scores on CamVid database as a function of the label patch size using Simple Fusion.

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 discarded all labels in the ground truth information when they were outside a radius of 24 pixels to a transition between two or more classes. This results in a drop to 41.9% of the original amount of labelled pixels. In Table 6.2, the corresponding results are listed when using a label neighborhood 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 our assumptions that the proposed framework yields to superior results, especially when classifying local label transitions of object classes.

6.4.2 MSRCv2 Database Experiments

To show that our method also yields to an improvement when the images are not entirely labelled, we performed another experiment on the MSRCv2 Database [Shotton et al., 2006]. This database consists of 532 images containing 21

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
Our method (Structure + Simple Fusion)	69.9	50.4	35.0
Our method (2-Full + Simple Fusion)	71.6	50.1	35.8
Our method (2-Full + Optimized Selection)	72.5	51.4	36.4

Table 6.2: Classification results for labelling transitions on the CamVid database for label patch size 11 × 11.

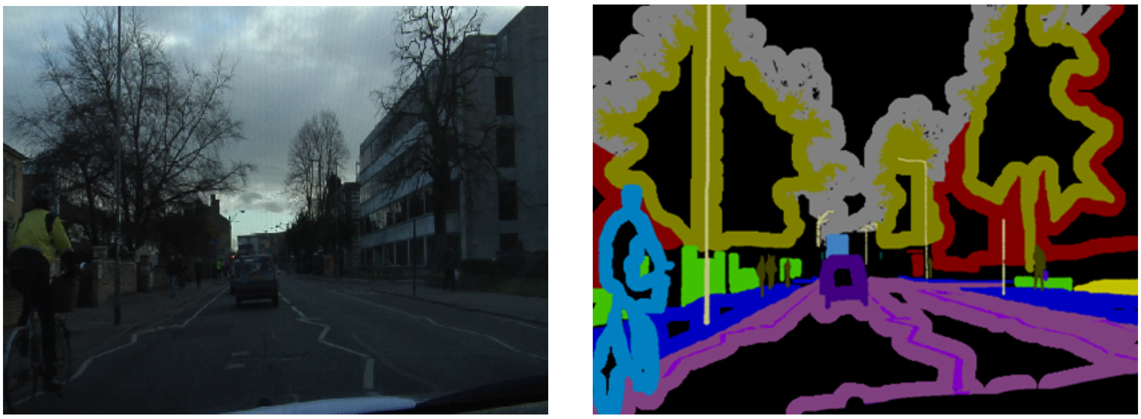


Figure 6.9: Training data examples used for experiments on labelling transitions experiment on the CamVid database. Left: Original image, right: Modified label image showing areas of considered label transitions.

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 contrast to the almost completely labelled CamVid database, the labellings for MSRCv2 are only available for 71.9% of the pixels, hence more roughly sketching the object classes of interest. In Figure 6.10 we show some qualitative results and in Table 6.3, we provide the scores for a label neighborhood 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.

6.4.3 KAIST Hanja2 Database

In our next experiment we demonstrate that our proposed method can also be used for reconstructing occluded regions in handwritten, Chinese characters of the

Method	*Global*	*Avg(Class)*	*Avg(Pascal)*
Texton forests naïve (supervised) [Shotton et al., 2008b]	49.7	34.5	-
Texton forests (Full system) [Shotton et al., 2008b]	72.0	67.0	-
RF using covariance features [Kluckner et al., 2009]	55.8	42.2	-
Our Baseline RF	54.8	43.4	28.3
Our Baseline RF + CRF	61.0	52.8	35.1
Our method (Structure + Simple Fusion)	60.8	51.0	33.8
Our method (2-Full + Simple Fusion)	60.8	51.1	33.9
Our method (2-Full + Optimized Selection)	63.9	55.6	37.6
Our method (2-Full + Optimized Selection) + Correlation coefficients	70.0	59.6	43.3

Table 6.3: Classification results on MSRCv2 database for label patch size 11 × 11.

KAIST Hanja2 Database. To this end, we reproduced an experiment from the work in [Nowozin et al., 2011], 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 considerably improved with structured class labels, as for the binary classification case they exhibit the local shape information of interest.

We used the original training (300 images) and testing data (100 images) of [Nowozin et al., 2011] and their respective, randomly generated occlusions for the *small* occlusion dataset. As a baseline, we obtain an average per-tree, pixel-wise classification result of 68.52% when evaluating 10 randomly trained decision trees (each with a maximum depth of 15). During training of the forest, we used 2000 iterations per node and simple pixel difference tests on the gray values, which were allowed to look at most 80 pixels away. We used the identical setup within our proposed structured class-label random trees, considering label patches of size 5 × 5.

In Table 6.4, we compare the classification results when using only our single, baseline decision tree, a tree ensemble of 10 trees and the Markov Random Field (MRF), the Decision Tree Field (DTF), the Regression Tree Field (RTF) (using regression trees with maximum depth 20) [Jancsary et al., 2012] results are taken from [Nowozin et al., 2011, Jancsary et al., 2012]. Additionally, we compare to our recent work in [Rota Bulò et al., 2012], where we introduced a new model denoted as ”Structured Local Predictors” (SLP). SLP are locally operating models, which provide a per-pixel labelling by exploiting contextual relations, learned from complex interactions between labels and a customizable intermediate representation of the image data. When using structured class labels as introduced in this work, we can boost the initial classification score of the single decision tree by almost 10%, outperforming sophisticated methods like MRF, DTF or RTF. The result is approximately on par with SLP, however, without the need for explicit parametrization of the pairwise interactions to be learned.

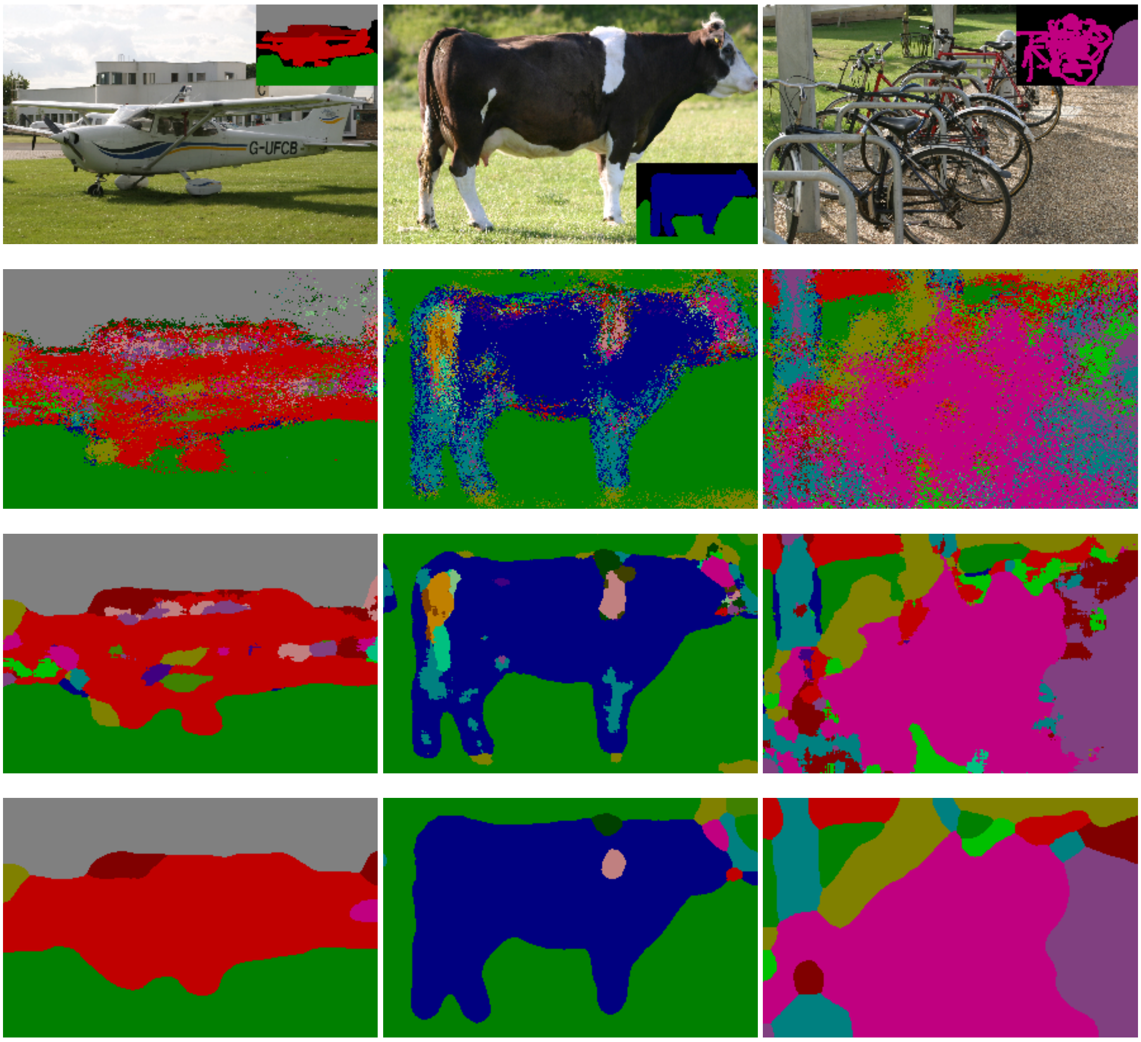


Figure 6.10: Qualitative labelling results on images of the MSRCv2 database. Top row: Original images with ground truth annotations. Second row: Labelling using our baseline random forest classifier. Third row: 2-Full + Simple Fusion. Last row: 2-Full + Optimized Selection. Best viewed in color.

6.4.4 Person Detection on TU Darmstadt Databases

To assess our proposed approach for the task of object detection as described in Section 6.3, 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, yielding to non-distinctive hypotheses in the Hough space. However, with our combined approach we also learn the local shape information by analyzing 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.

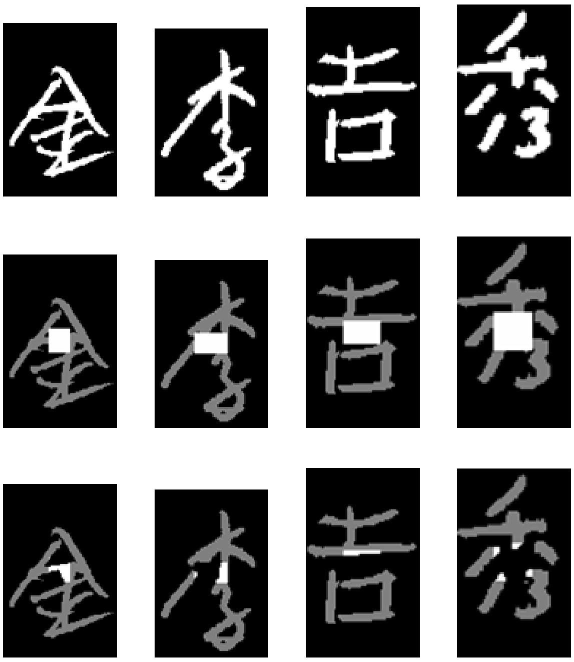


Figure 6.11: Example reconstructions for experiment on occluded Chinese Characters. Top row: Ground truth images. Center row: Test images with occlusions. Bottom row: Restored characters using our method.

Model		Global classification score [%]
Single Decision Tree (On average)		68.52
Entire Forest (10 trees)		74.95
Markov Random Field (MRF)	[Nowozin et al., 2011]	75.18
Decision Tree Field (DTF)	[Nowozin et al., 2011]	76.01
Regression Tree Field 1D (RTF 1D)	[Jancsary et al., 2012]	76.39
Regression Tree Field 2D (RTF 2D)	[Jancsary et al., 2012]	77.55
Structured Local Predictors (SLP)	[Rota Bulò et al., 2012]	78.07
Our method	5 × 5 label patch	**78.09**

Table 6.4: Reconstruction results for KAIST Hanja2 dataset in terms of global classification score in [%] for occluded regions.

We evaluated our method on the TUD pedestrian databases [Andriluka et al., 2008], 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 [Riemenschneider et al., 2012], providing a total number of 1216 bounding boxes. Please note that this annotation is even more detailed than the one presented in [Barinova et al., 2010] with 1018 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 $2 \cdot 400 \cdot 2 \cdot 125 = 200k$ training samples per tree.

The most related work and therefore also the baseline in our experiments is the Hough Forest [Gall and Lempitsky, 2009]. To guarantee a fair comparison, we used the same training parameters for [Gall and Lempitsky, 2009] and our enhanced, structured class-label Hough forest: We trained 20 trees and the training data (including horizontally flipped images) was sampled homogeneously per category per image. The patch size was fixed to 20×20 and we performed 1600 node tests for finding 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 [Gall and Lempitsky, 2009]. In order to obtain the object detection hypotheses from the Hough space, we use the same Non-maximum suppression (NMS) technique in all our experiments as suggested in [Gall and Lempitsky, 2009]. 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 $\geq 50\%$.

For additional comparisons, we provide the results presented in the recent work on joint object detection and segmentation of [Riemenschneider et al., 2012], from which we also provide evaluation results of the Implicit Shape Model (ISM) [Leibe et al., 2008]. Please note that the results of [Riemenschneider et al., 2012] are based on a different baseline implementation. Additionally, we include the results obtained with the publicly available code for the approach of [Barinova et al., 2010]. Finally, we also list the scores obtained in our recent work [Kontschieder et al., 2012] being trained according to the same training protocol and parameters as for our structured class-label Hough forest.

6.4.5 Evaluation on Campus scene

First, we discuss the results obtained on the Campus scene. This data set consists of 71 images showing walking pedestrians at severe scale differences and partial occlusions. The ground truth we use has been released with [Barinova et al., 2010] and contains a total number of 314 pedestrians. Figure 6.12 shows precision/recall curves when evaluating on 3 scales (factors 0.3, 0.4, 0.55) on the left and using 5 scale factors (0.34, 0.42, 0.51, 0.65, 0.76) on the right.

When considering 3 scales and a precision rate of 90%, we obtain an considerable improvement of $\approx 8\%$ over the Hough forest baseline implementation. Moreover, we also improve on our recently presented results in [Kontschieder et al., 2012] by $\approx 2\%$. When using 5 scales we get even better results, i.e. we improve by $\approx 16\%$ over the baseline Hough forest and still $\approx 5\%$ over our earlier work [Kontschieder et al., 2012].

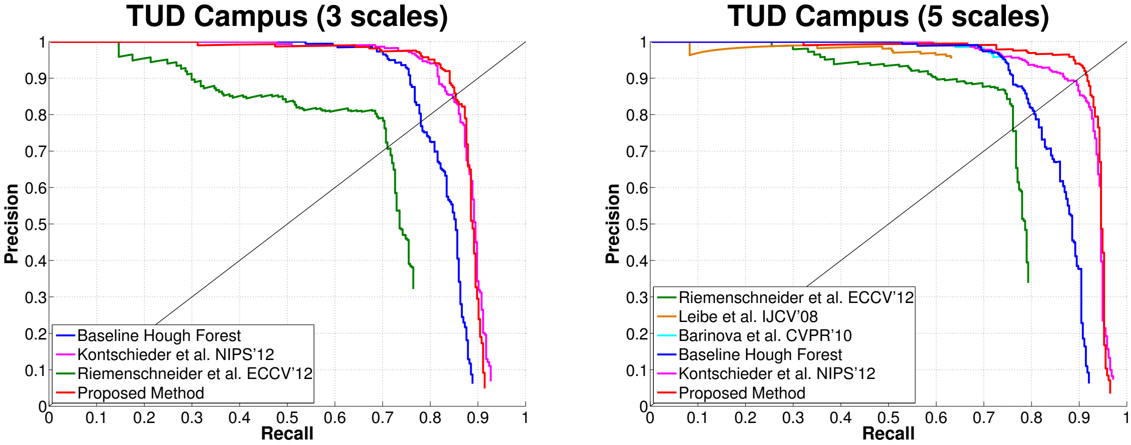


Figure 6.12: Precision-Recall curves for pedestrian detection experiments on TUD Campus scene.

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 5 scales yield to higher detection rates than 3 scales. In fact, appearance and scale should of course be as close to the ones observed during learning.

Even though the improvement over [Kontschieder et al., 2012] is smaller, we stress that in the proposed approach we do not explicitly learn from nearby object entities as it is possible in [Kontschieder et al., 2012]. However, such object-instance specific information might also be included in here.

The visualizations in Figure 6.13 show exemplary results of our method. Yellow dashed bounding boxes indicate ground truth and green solid boxes are our detected hypotheses. Please note the high localization accuracy of the detected objects and especially. In the second row we show false positive detections of our approach (cyan bounding boxes) when using the most recent ground truth annotations of [Barinova et al., 2010]. Actually, all provided illustrations contain true locations of pedestrians, however with significant occlusion.

6.4.6 Evaluation on Crossing scene

Figure 6.15 shows the results when the same forests are tested on the Crossing scene, using the ground truth annotations of [Riemenschneider et al., 2012]. The data set shows walking pedestrians (Figure 6.14) 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 ($\approx 2\%$ gain at a precision of 90%) and are approximately on par

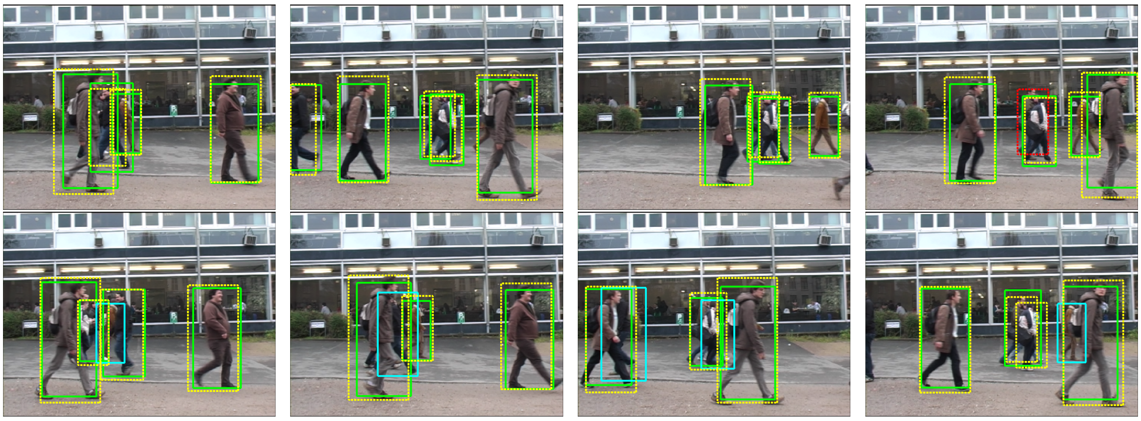


Figure 6.13: Examplary detection results on Campus scene (evaluation with 5 scales). Bounding box colors: Yellow dashed for ground truth, green for obtained detections and red for missed detections. The second row shows false positive detections of our classifier (in cyan) for given ground truth annotations. Please note the actual presence of persons at hypothesized locations.

with our earlier work in [Kontschieder et al., 2012]. Please note that the training data only contains single object instances, i.e. each training image contains exactly one annotated person. In such a way, mutual occlusions cannot directly be learned from this data.

As in the previously shown qualitative results on the Campus scene we show ground truth annotations in yellow dashed bounding boxes, our obtained detections in green and missed detections in red boxes, respectively. The illustrations in Figure 6.14 show how we are able to accurately outline the respective object locations, even in severely occluded areas. The provided failure cases in the last two images illustrate that the annotations are very strict and contain also highly occluded objects, making it a very challenging database.

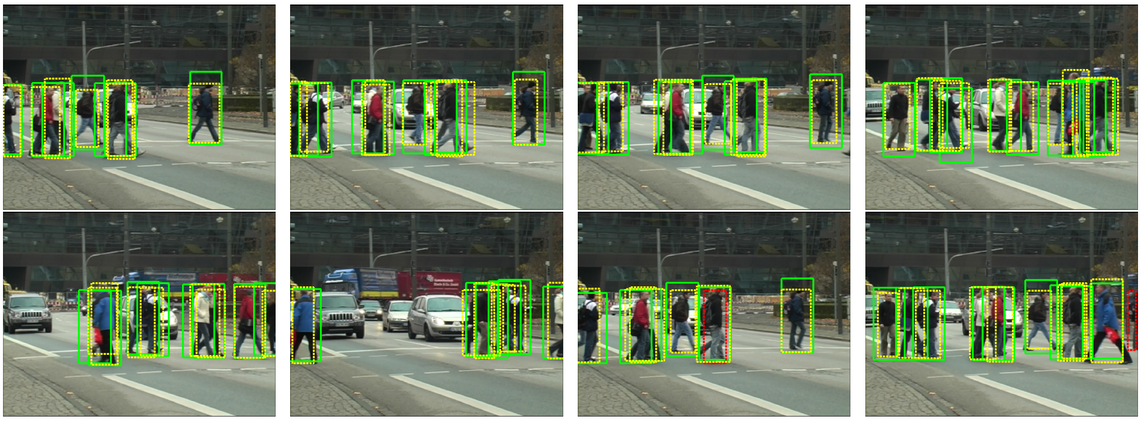


Figure 6.14: Exemplary detection results on Crossing scene. Bounding box colors: Yellow dashed for ground truth, green for obtained detections and red for missed detections.

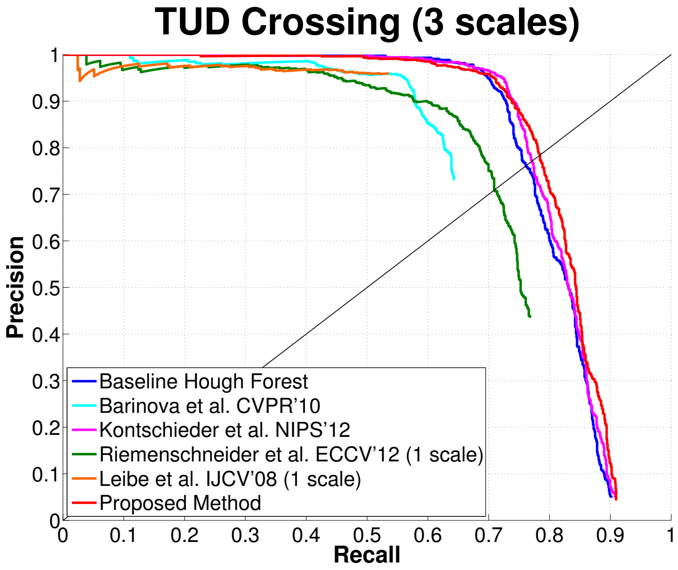


Figure 6.15: Precision-Recall curves for pedestrian detection experiments on TUD Crossing scene when evaluating three scales.

6.5 Additional Discussion of Proposed Approach

One of the key contributions of our proposed approach is the way we process additional information from the label space, that is inherently available in the ground truth information at training time. Mainly, we are able to capture plausible, local label configurations and information about the shape that separates them. In this sense, we can argue that standard random forests are a special instance of our proposed learning method, which only uses an atomic label neighborhood of size *one*.

When considering such an extended label neighborhood, the training process can be driven in a way to account for k-label joint distributions, helping to better explore the additional label information while trees are grown from the root to the respective leaf nodes. An interesting observation resulting from this learning approach is visualized in Figure 6.16, where we clearly see that there is high correlation among collected samples in both, the input (or feature) space *and* the output space, which cannot be explicitly forced in the training of standard classification trees. In other words, we can monitor high similarities on the feature space but also on the label space regarding the topological distribution of class labels.

This additional information can now be elegantly exploited for multiple purposes, as demonstrated in our experiments: We start with discussing the effect for the task of semantic image labelling. The use of structured label patches allows to perform more robust labellings as each pixel in the label patch gets classified also from its neighborhood (and may in turn classify its own neighborhood). This is helpful in the case when borders between labels shall be continued, but also to compensate high intra-class variabilities (e.g.

strong appearance changes) and clutter, typically leading to noisy labellings. In the latter case, structured label patches have a smoothing effect in the label space which yields to more consistent labellings. Please note that such smoothing effects can also be interpreted in the light of the pairwise terms of a standard MRF energy formulation, encouraging neighboring pixels to take on the same labels such that appropriately segmented regions are obtained. However, in our case we will not propagate implausible label configurations, as the structured label patches will only contain what is provided in the ground truth information.

As a special, binary labelling case, we have conducted an experiment on Chinese character reconstruction where essentially the calligraphy (strokes, character outline and type) in occluded regions has to be restored (see Section 6.4.3). Please note that for this task the local shape development is of special interest as it captures many calligraphic properties. However, here we show an additional benefit of our method arising from the optimized label patch selection introduced in Section 6.2.5. Since the reconstruction task is only performed in the occluded areas, our proposed selection mechanism can optimally select from the provided label patch candidates of the forest in the sense the agreement function is designed in Equation (6.5). Another interpretation is that the surrounding, non-occluded regions in the image form an initial solution which is guiding the reconstruction process in the occluded ones, given the *alphabet* provided by the trees.

When enhancing our approach to account also for the object detection problem as described in Section 6.3, we can observe a clear improvement over the Hough forest [Gall et al., 2011], i.e. the most related work which we have also taken as a baseline. In our experiments on pedestrian detection we find similar behavior in the leaf nodes of the trees as mentioned for the labelling task: Distinguished parts (in this case along the silhouette) of the human body like legs, arms and head are better clustered in both, the feature and label space, leading to more concentrated center votes. Consequently, votes with less variance produce more distinctive peaks in the Hough voting space and in turn can be better extracted in the subsequent non-maximum suppression stage. In our experiments we also find favorable behaviour with respect to the recent work in [Kontschieder et al., 2012] which also exploits context in random forests, but in a different way: There, the context needs to be consequently updated and analyzed in the way the feature space is exploited, while the proposed approach can even discard the label information once the training process is finished. In this sense, our method is computationally much more efficient than [Kontschieder et al., 2012] and maintains the same computational complexity as the Hough forest during test time. Please note that the work in [Kontschieder et al., 2012] offers several extensions which may be applied to the proposed work as well, e.g. the way the the quality of the node split is evaluated happens according to a different objective function, it provides a way to simultaneously learn from multiple training instances of the same category and the authors show a way to isolate individual modes of the Hough votes.

Finally, we conclude by mentioning that the major drawback of our method is probably

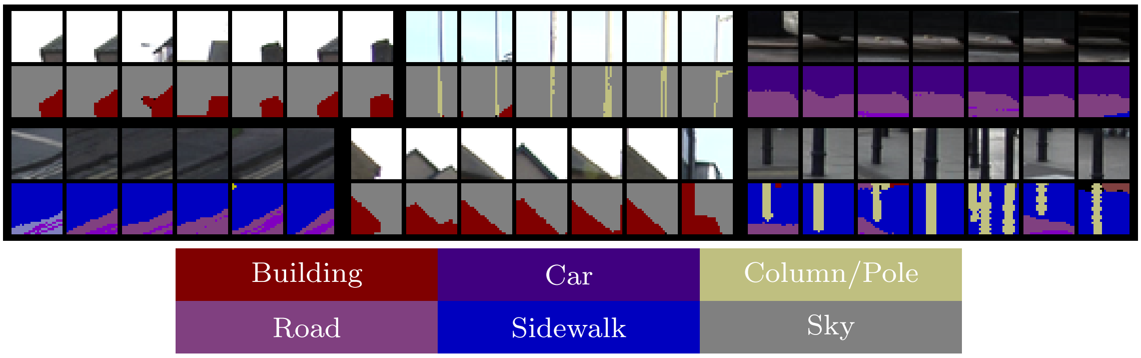


Figure 6.16: Illustration of feature patches with corresponding label patches, collected from different leaf nodes when trained on CamVid database. Bottom rows: Label sets and associated colors. Best viewed in color.

the need for densely labelled training data. However, this problem is shared with state-of-the-art image labelling algorithms and the results of our experiments on the MSRCv2 database indicate that also rough labellings are well handled by our method.

6.6 Conclusion

In this chapter 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 local label neighborhood in the training process and therefore intuitively learned valid labelling transitions among adjacent object categories. During the tree construction, we used joint label statistics of the training data in the node split functions for exploring the structured label space. For classification, we provided two possibilities for fusing the structured label predictions: A simple method using overlapping predictions and a more principled approach, selecting most compatible label patches in the neighborhood. 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.

Chapter 7

Geodesic Forests for Learning Coupled Predictors

Contents

Conventional random forest based methods for image labelling tasks like object segmentation make predictions for each variable (pixel) independently. This prevents them from enforcing dependencies between variables and translates into locally inconsistent pixel labellings. On the other hand, random field models or the semantic segmentation approach presented in the previous chapter encourage spatial consistency of labels, however, at increased computational expense during inference. This chapter presents a novel and efficient forest based model that achieves spatially consistent image segmentation by encoding variable dependencies in the feature space the forests operate on. Such correlations are captured via new long-range, soft connectivity features, computed efficiently via generalized geodesic distance transforms. Our model can be seen as a generalization of the Semantic Texton Forest, Auto-Context, and Entangled Forest models that have produced convincing results on segmentation problems. Another extension of the standard classification forest model presented in this chapter is the development of a novel objective for training decision forests, encouraging predictions to be consistent with spatial context. Our model is validated on the task of semantic image segmentation on four diverse image datasets and compared to the approach presented in the previous chapter. Experimental results show that the presented model improves over state-of-the-art forest based models and also over the conventional pairwise CRF models for many datasets.

7.1 Introduction

Many problems in computer vision can be formulated in terms of structured output prediction. Here, the term ”structured” relates to the presence of dependencies between output variables. For instance, in image labelling problems such as object segmentation or image denoising, the variables associated with neighboring pixels are more probable to take the same labels. In recent years, random forests have become very popular for the solution of a wide variety of image labelling problems - from organ segmentation in medical images [Montillo et al., 2011] and semantic segmentation [Shotton et al., 2006, Shotton et al., 2008b] to human pose estimation for KINECT [Shotton et al., 2012].

There are a number of reasons for the success of forest models, including scalability to large amount of data, ability to learn long-range dependencies between features and output variables, little tendencies to overfitting and finally, extreme efficiency in making predictions. The last of these qualities is derived from the independence assumption made by these methods. In fact, conventional decision forests ignore the structure in output spaces and make predictions for each output variable independently. This assumption prevents them from enforcing dependencies between variables, and for image segmentation problems, translates into pixel labellings that do not follow object boundaries and are inconsistent with spatial context.

To overcome these problems, Markov or Conditional random fields (MRF/CRF) [Carbonetto et al., 2004, Lafferty et al., 2001, Blake et al., 2011] are used as a post-processing step [Nowozin et al., 2011, Shotton et al., 2006]. For instance, in [He et al., 2004a, Shotton et al., 2006] image segmentation is achieved by first computing pixel-wise unaries via supervised classification, and then smoothing the labels with a CRF. The more recent works in [Nowozin et al., 2011, Jancsary et al., 2012] essentially present a CRF model, where the pairwise potentials (and not just the unaries) are conditioned on the data and predicted via a single tree. Another way of mixing trees and random field models is presented in [Payet and Todorovic, 2012], where again, the underlying model is a CRF. Alternatively, with our random forest approach described in the previous chapter, we obtain spatial smoothness by combining structured class-labels that are learned by incorporating joint statistics from the label space neighborhood. Although all of these approaches lead to better results, this comes at the cost of increased computational expense at test time.

What we present here is a computationally efficient random forest model for structured output prediction. Our framework overcomes the above mentioned problem by incorporating learned spatial context directly *within* the forest itself. This leads to smooth and correlated, pixel-wise labellings without the need for random-field-based post-processing. Our framework achieves this by encoding variable-dependencies in the features where the forest operates on. Such correlations are captured via a new type of long-range, soft connectivity features which can be efficiently computed using generalized geodesic distance transforms in the spatial and feature spaces. Another change with respect to the standard classification forest is a novel training objective that encourages the trees to make predictions that are consistent with local context. We demonstrate the effectiveness of our model on the task of segmenting four diverse image datasets: face images, medical scans, depth images and driving videos of the CamVid dataset. Quantitative results demonstrate the superiority of our model both in terms of accuracy and efficiency, with respect to state-of-the-art forest based models and also conventional, grid-based pairwise CRF.

Our work is related to methods based on sequential classification. The recent work on auto-context [Shotton, 2007, Tu and Bai, 2010], stacking [Munoz et al., 2010, Wolpert, 1992] and entanglement [Montillo et al., 2011] has shown how a sequence of classifiers using the output of the previous classifier as input to the next can both, effectively capture spatial context (e.g. from medical torso images it can learn that the heart is between the lungs) and improve accuracy. In [Fröhlich et al., 2012], the relationship between anytime classification and intermediate predictions within decision trees is shown. In [Ross et al., 2011] the authors show how conventional message-passing inference on graphical models may be interpreted as a sequential probabilistic inference algorithm.

Our model can be seen as a generalization of Semantic Texton Forest [Shotton et al., 2008a], Auto-context [Shotton, 2007, Tu and Bai, 2010], and Entanglement forests models [Montillo et al., 2011]. In fact, our algorithm builds upon these models by using:

- long-range soft connectivity features that encode variable-dependencies and result in consistent pixel-label predictions
- an objective for training forest models, which is inspired from the energy functions used in random field models, and encourages our model to make predictions consistent with local context.

7.2 Background and Problem Formulation

In addition to the notation defined in Chapter 2, we provide additional variables and their symbols used in this chapter. The input or feature domain is denoted with $\mathcal{X}$ and the output or label domain is denoted with $\mathcal{Y}$. An image I is a multi-channel matrix mapping pixels $\mathbf{u}$ (i.e. elements of $\mathbb{Z}^2$) to m-dimensional feature vectors $\mathbf{v}(\mathbf{u}) = \{v_1, \ldots, v_m\} \in \mathbb{R}^m$. We cast the semantic image segmentation problem as that of associating each pixel location $\mathbf{u}$ with a discrete, categorical label c_l from the output space $\mathcal{Y} = \{c_1, \ldots, c_C\}$.

As we deal with a classic supervised classification problem we assume to be provided with a set of labelled training images $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ from where we take the set of training samples $\mathcal{Z} \subseteq \mathcal{D}$. Each individual sample $z_i = \{x_i, c_i\}$ comprises of a sample $x_i = (\mathbf{u}, I)$ taken in the feature space $\mathcal{X}$ and its associated label c_i.

Let $\mathbf{y} = \{y_\mathbf{u} | \mathbf{u} \in \mathbb{Z}^2\}$ denote the vector of variables $y_\mathbf{u}$ predicted by our classifier for pixel $\mathbf{u}$. In tree-based classifiers we use $\mathbf{y}_d$ to denote predictions obtained at depth d in the tree. In a forest, D denotes the maximum tree depth and $T = |\mathcal{T}|$ is the number of trees in the ensemble $\mathcal{T}$.

Random field models. Given an image I, its most probable labelling can be inferred by finding the most probable solution under the posterior distribution:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} P(\mathbf{y}|I) = \arg\max_{\mathbf{y}} P(I|\mathbf{y})P(\mathbf{y}) \tag{7.1}$$

The conventional pairwise random field models assume that the posterior distribution factorizes into a product of unary and pairwise potential functions as:

$$P(\mathbf{y}|I) = \prod_{\mathbf{u}} \psi(y_\mathbf{u}, \mathbf{v}(\mathbf{u})) \prod_{(\mathbf{u},\mathbf{q}) \in \mathcal{N}} \psi'(y_\mathbf{u}, y_\mathbf{q}, \mathbf{v}(\mathbf{u}), \mathbf{v}(\mathbf{q})) \tag{7.2}$$

where the set $\mathcal{N}$ of pixel pairs describes a neighbourhood system and is pre-defined. Although this factorization assumption makes inference of the Maximum a Posteriori (MAP) solution for many models tractable, it severely limits the expressive power of the model. Further, inference and learning are quite computationally expensive.

Decision forest models. Random decision forests, on the other hand, assume that the posterior decomposes over individual variables as: $P(\mathbf{y}|I) = \prod_\mathbf{u} \varphi(y_\mathbf{u}, \mathbf{v}(\mathbf{u}))$. This factorization enables the model to ignore the dependency between output variables and makes predictions independently and efficiently.

Predictions with decision forests. To make predictions, a series of feature tests starting at the root node are applied to each pixel (i.e. patch) independently. For instance, at the root node, a test is computed on the feature response for pixel $\mathbf{u}$ and, depending on the results, the prediction process moves on to the left or right branch of the tree. The procedure is repeated until the pixel reaches a leaf node. At this point the empirical class distribution $\phi(y_\mathbf{u}|\mathbf{v}(\mathbf{u}))$ associated with the leaf is read off. The MAP label for the pixel is obtained as:

$$y^*_\mathbf{u} = \arg\max_{y_\mathbf{u}} \varphi(y_\mathbf{u}, \mathbf{v}(\mathbf{u})). \tag{7.3}$$

Forest training. Training involves: i) selecting the feature tests at split nodes of the tree, and ii) estimating the empirical class distributions $\varphi(y_\mathbf{u}|\mathbf{v}(\mathbf{u}))$ associated with each leaf node. Traditionally, the structure of a decision tree is learned in a greedy fashion where

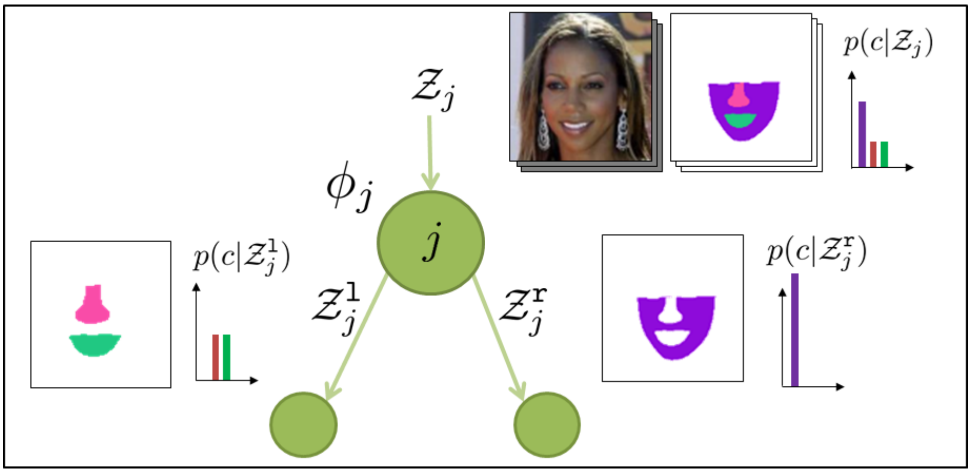

$\mathcal{Z}_j$
ϕ_j
j
$p(c|\mathcal{Z}_j)$
$p(c|\mathcal{Z}_j^{\mathtt{l}})$
$\mathcal{Z}_j^{\mathtt{l}}$
$\mathcal{Z}_j^{\mathtt{r}}$
$p(c|\mathcal{Z}_j^{\mathtt{r}})$

Figure 7.1: **Optimization procedure in a split node j of a tree.** Split node training seeks a node parametrization ϕ_j which maximizes the information gain (class purity) as well as spatially compact pixel clusters.

for each internal node j in the tree the split parameters ϕ_j (see Fig. 7.1) that lead to greater purity of class labels in the child nodes (greater information gain) are selected. In a decision tree hierarchy, we assume split nodes to be indexed in breadth-first order using integers j, where $j = 0$ corresponding to the root node. During training we have a different subset $\mathcal{Z}_j$ of training data associated with a different node j. Thus, we denote the initial training set available to the root node with $\mathcal{Z}_0$.

7.2.1 Coupling Forest Predictions - Revealing Hidden Correlations

Although the independence assumption enables efficient training and rapid predictions with random forests, it prevents the model from enforcing dependencies between variables which for image segmentation problems, translates into pixel labellings that do not follow object boundaries and are not consistent with local or global context. In the work presented in this chapter, we overcome this problem and encourage forests to produce spatially compact/coherent pixel labellings. In what follows, we will show how a learned model of spatial context can be encoded *within* a decision forest framework. This leads to smooth, pixel-wise image labellings without the need for post-processing steps.

One of the key theoretical insights of our work is the observation that although forests make predictions for each variable independently, these predictions are related due to correlations at the feature level. For instance, consider a semantic image segmentation task and the class predictions of two pixels $\mathbf{u}$ and $\mathbf{q}$. From (7.3) it is easy to see that the MAP labels $y^*_{\mathbf{u}}$ and $y^*_{\mathbf{q}}$ are functions of the features responses $\mathbf{v}(\mathbf{u})$ and $\mathbf{v}(\mathbf{q})$ i.e. $y^*_{\mathbf{u}} = f(\mathbf{v}(\mathbf{u}))$ and $y^*_{\mathbf{q}} = f(\mathbf{v}(\mathbf{q}))$. This relationship implies that output-variable dependencies can be encoded in the features that the forest operates on. We exploit this insight to couple forest predictions in two ways:

- We enable long-range geodesic features for soft connectivity between image regions
- We train entangled classification forests, where geodesically smoothed, intermediate class posteriors estimated at higher levels in each tree influence the training of the tree lower levels.

We describe details of these two contributions in the next three sections.

7.3 Long-range, soft connectivity features

The need for long-range connectivity features. In [Lepetit and Fua, 2006, Shotton et al., 2012, Winn and Shotton, 2006] the authors have shown how simple pixel comparison features can be effective in classification tasks when used within a decision forest. Such features are extremely fast to compute (they involve just pixel-wise read-outs), but not particularly expressive. This is illustrated in Figure 7.3 where we compare simple pair-wise intensity difference features (left image) with a feature response based on evaluating the geodesic path connecting the point pair (right image). Shortest-path based features should be able to capture the idea of connectivity between points which could be used within a learned segmentation algorithm to decide whether two points should have the same class label or not. For example, the points r_3 and p_3 have identical intensity values. However, one is in the lungs and the other in the air outside the body. Since the shortest path connecting them has a high geodesic length (because it has to cross high image gradients) this provides some evidence that the two points are not part of the same object/class. Similarly, the points r_2 and p_2, despite being far from each other in terms of Euclidean distance, they are close in geodesic terms. This provides evidence that they belong to the same object (the aorta, an elongated tubular object, in this case). Of course, using this signal directly would be too fragile and we need to find an efficient way of integrating it within a learning framework.

In theory these geodesic-based visual features could capture edge-aware spatial smoothing, similar to CRF. However, these features need to be available at test time, for *any* pair of pixels. But computing any-pair shortest paths within an image on the fly is infeasible. We circumvent this problem by proposing a novel set of visual features which are computationally efficient and yet manage to capture the degree of connectivity between probabilistically defined image regions. They are based on the use of *generalized* geodesic distances, as introduced in [Criminisi et al., 2008, Criminisi et al., 2011] and summarized next for completeness.

Generalized geodesic distances. Given a grey-valued image I, and a real-valued object "soft mask" $M(\mathbf{u}) : \Psi \in \mathbb{N}^d \to [0,1]$, weighted by an importance parameter ν, the generalized geodesic distance Q is defined as follows:

$$Q(\mathbf{u}; M, \nabla I) = \min_{\mathbf{u}' \in \Psi} \left(\delta(\mathbf{u}, \mathbf{u}') + \nu M(\mathbf{u}')\right) \tag{7.4}$$

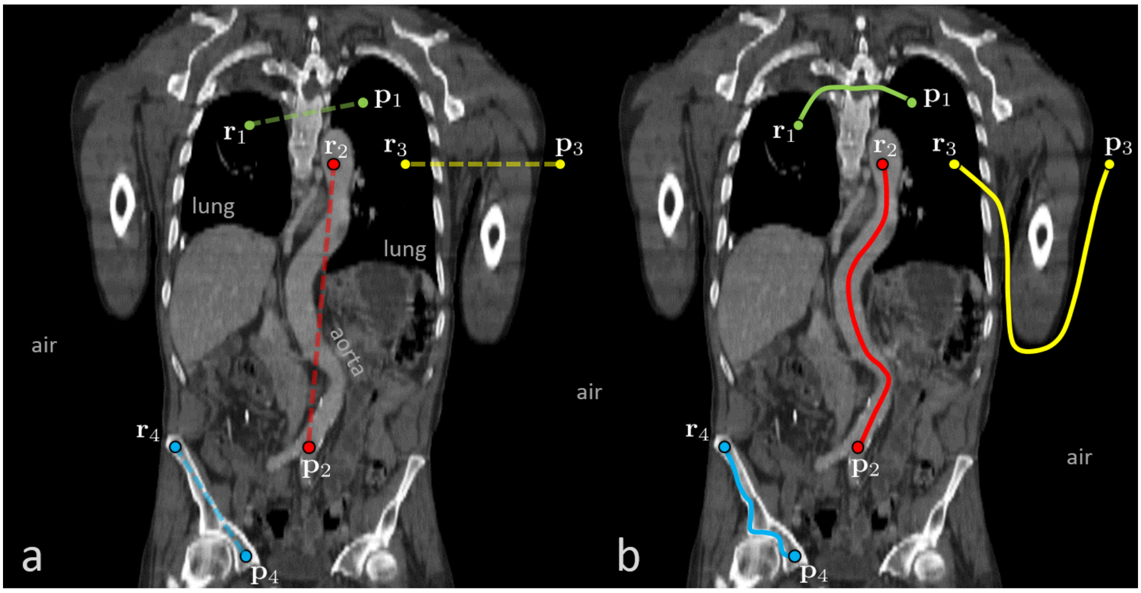


Figure 7.2: **Connectivity features.** A 2D vertical slice through a 3D computed tomography (CT) scan. **(a)** Commonly used feature responses are computed e.g. by looking at the difference of intensities between pairs of pixels. These features ignore what happens in between the two pixel positions. **(b)** Given a pair of pixels, computing the cost of the shortest path connecting them carries richer information. Thus, for instance the points r_3 (within the lungs) and p_3 (outside the body), despite having identical intensity, the shortest connecting path has to go through regions of high gradient. This is an indication that the two points belong to different classes. However, computing *all-pairs* of shortest paths is clearly not feasible. See text for our proposed, efficient solution.

with the geodesic distance between two points $\mathbf{u}$ and $\mathbf{q}$ defined as:

$$\delta(\mathbf{u},\mathbf{q}) = \inf_{\Gamma \in \mathcal{P}_{\mathbf{u},\mathbf{q}}} \int_0^{l(\Gamma)} \sqrt{1+\gamma^2(\nabla J(s)\cdot \Gamma'(s))^2} ds\,. \tag{7.5}$$

Γ is a path connecting the two points and $\Gamma'(s) = \frac{\partial \Gamma}{\partial s}$ is the spatial derivative. $\mathcal{P}_{\mathbf{u},\mathbf{q}}$ is the set of all possible paths, $l\,(\Gamma)$ is the length of the path and $\gamma \geq 0$ is a weighting parameter controlling the influence of the image gradients. When setting $\gamma = 0$, (7.5) coincides with the Euclidean distance, as the minimal distance is a straight line connecting the points $\mathbf{u}$ and $\mathbf{q}$. Consequently, (7.4) defines the distance of any point in the image to a region in the image defined by the weighted "soft belief" mask M.

Soft connectivity to a class region. Furthermore, we assume that we have an image I and also the belief region M associated with a chosen class. Now we can compute the distance of every point in the image to the given class region. Note that the class region is defined in a soft way and we do not need to select hard seed positions. In the

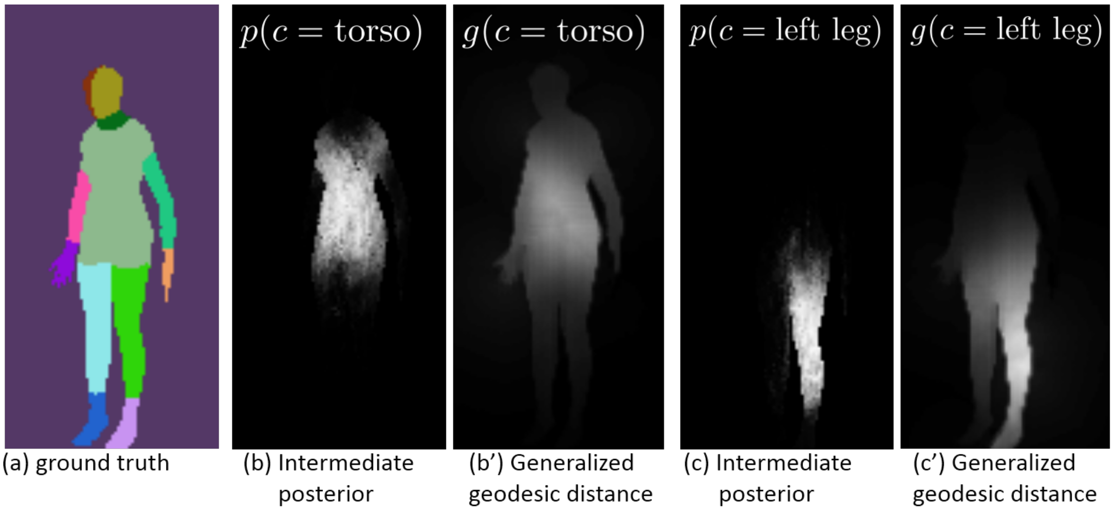

$p(c = \text{torso})$
$g(c = \text{torso})$
$p(c = \text{left leg})$
$g(c = \text{left leg})$
(a) ground truth
(b) Intermediate posterior
(b′) Generalized geodesic distance
(c) Intermediate posterior
(c′) Generalized geodesic distance

Figure 7.3: **Generalized geodesic distances from probabilistic class regions. (a)** Ground truth body part labels for a depth image. **(b, c)** Approximate class probability maps $p(c|\mathbf{u})$; assumed given here. **(b', c')** Geodesic-smoothed probability maps $g(c|\mathbf{u})$. Notice how the function g can be interpreted as a diffused version of the noisier probabilities p, where contast sensitivity is modulated by the geodesic weight γ in (7.5). The visual features used in our forest model are simple pixel read-outs of the g maps. Thus, while maintaining great efficiency, they capture long-range connectivity (of a pixel to a class region) information.

existing literature distance transforms have been used for manually assisted segmentation only (e.g. in [Gulshan et al., 2010] and [Criminisi et al., 2008]). Going from interactive to automatic segmentation is challenging because of the seed selection problem, and runtime efficiency issues.

The belief regions can be the output of a given, probabilistic classifier[1]. We can think of having C such masks and thus C such distances associated with each input image.

To make this point clearer, Figure 7.3 shows an illustration. Given a depth image (e.g. acquired with a laser range finder, or with Kinect), we assume we have a probabilistic classifier which when tested produces the posterior probabilities $p(c = \text{torso})$ and $p(c = \text{left leg})$. We can use those probabilities as our soft masks M for the generalized geodesic distance transform and the corresponding distances will result as $g(c = \text{torso})$ and $g(c = \text{left leg})$. Note how the g maps look like a smoothed version of the class probabilities p. Contrast sensitivity is modulated by the parameter $\gamma \geq 0$ in (7.5).

In the next section we will show how generalized geodesic distances can be used effectively as long-range visual features via an entangled decision forest framework. Note that we are not really solving the problem of computing all pairs of shortest paths, in fact, it

[1] e.g. for example, the map M could be given by the probability of a pixel belonging to the class sky or pedestrian in an image.

turns out that we do not need to. As it will become clearer later, computing only C generalized geodesic distances per image (i.e. one per object category of interest) is sufficient to generate discriminative, long-range connectivity features. Moreover, generalized geodesic distance transforms can be computed in linear complexity with respect to the number of pixels [Criminisi et al., 2011], using constant-time complexity lookup-table operations.

7.4 Entangled geodesic forests and their features

Here we are interested in extremely efficient semantic segmentation. Thus, we use a variant of decision forests, because of their speed and flexibility [Amit and Geman, 1997, Breiman, 2001, Criminisi and Shotton, 2013]. In what follows, we describe our extension to obtain coherent segmentations.

Figure 7.4 gives an illustration of what was introduced in the work to entangled forests [Montillo et al., 2011] and is also exploited in this work. All trees are trained

- in parallel
- in breadth-first order
- in sections.

When training the first section (section 0) only *appearance-based features* (e.g. raw intensities) are available. However, when training the next section more *derived* features become available. In fact, the output class posteriors $p(c|I)$ of the previous section may be used as input to the next [Montillo et al., 2011, Kontschieder et al., 2012]. Here, we further augment such features by using the geodesically smoothed versions of the posteriors, $g(c|I)$ (see Figure 7.3). In this sense, we can use connectivity features computed with efficient geodesic transforms to couple output predictions of the forest for modelling long-range semantic context directly within the feature space of the classifiers.

More formally, we are given an ordered set of sections $(s_0, s_1, \ldots, D)$, where s_i indicates the maximum depth of the i^{th} section and D is the maximum tree depth. Given a class posterior $p_{s_i}(c|\mathbf{u})$ computed at the i^{th} section (with $i > 0$), its geodesically smoothed version is defined as

$$g_{s_i}\left(c|\mathbf{v}(\mathbf{u})\right) = \frac{1}{W}\, p_{s_i}(c|\mathbf{v}(\mathbf{u}))\; e^{-\dfrac{Q\left(\mathbf{u}; p_{s_i}(c|I), \nabla I\right)^2}{\sigma^2}} \tag{7.6}$$

with W a normalization factor to ensure probabilistic normalization: $\sum_c g_{s_i}(c|\mathbf{u}) = 1$ and $Q(\cdot)$ as defined in (7.4).

Feature responses for a reference pixel $\mathbf{r}$ are defined as a function of tree depth d, and as sum, differences or absolute differences between two pixel probe values in different channels, i.e. $v_i^d(\mathbf{r}) = F_k^d(\mathbf{u}_1) + F_k^d(\mathbf{u}_2)$, $v_i^d(\mathbf{r}) = F_k^d(\mathbf{u}_1) - F_k^d(\mathbf{u}_2)$ or $v_i^d(\mathbf{r}) = |F_k^d(\mathbf{u}_1) - F_k^d(\mathbf{u}_2)|$ where $k \in \{0, 1, 2\}$ denotes the *channel* where features are computed, and:

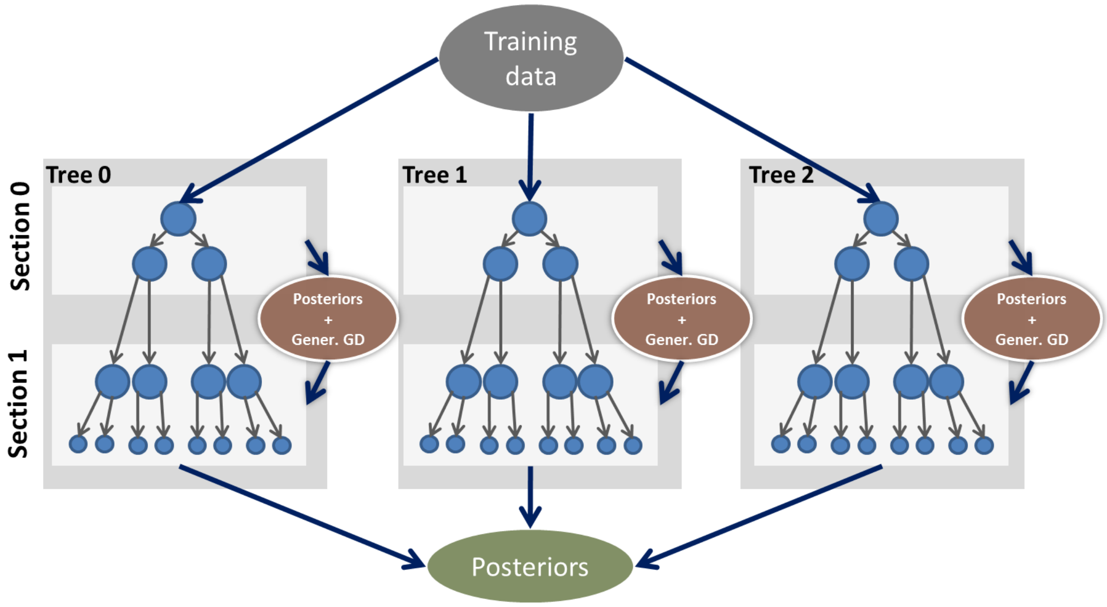


Figure 7.4: **An entangled geodesic forest.** A forest with three entangled trees. The trees are entangled because intermediate predictions of the top section are used (together with raw intensity features) as features for the training of the bottom section. There is only one entanglement section in this example.

- $F_0^d(\mathbf{u}) = I(\mathbf{u})$, i.e. the raw image intensities,
- $F_1^d(\mathbf{u}) = p_{s(d)}(c|(\mathbf{u}))$, i.e. the intermediate forest posteriors computed in the section $s(d)$ defined by the depth d, and
- $F_2^d(\mathbf{u}) = g_{s(d)}(c|(\mathbf{u}))$, i.e. the geodesic-filtered posteriors, capturing connectivity of point $\mathbf{u}$ to the seed region defined for class c.

The entangled feature channels $k = 1, 2$ are available only for section s_1 and greater. Furthermore, they can be computed very efficiently since they amount to little more than table look-ups.

7.5 Random Field inspired Forest Training Loss Functions

This section describes the second main contribution of this chapter which constitutes of a new loss function for the forest training procedure. In what follows, we depart from the traditional information-theoretic training process typically used in classification forests and derive a random-field inspired loss function. Moreover, we analyze both training functions with respect to an important training-data dependent circumstance, i.e. whether the training data is homogeneously distributed among the target categories (=balanced) or not (=imbalanced). Despite one can always sample from the training set in a way that the balance criterion is fulfilled, this might result in suboptimal use of the available training data where much available information is left unused.

7.5.1 Balanced classes

Implicitly, this case is also what we have introduced and assumed in the general notation and terminology of classification trees in Chapter 2.6.1 on page 16, using information gain as split node criterion. Here, we will walk through the maths in a more detailed way and introduce a novel, random field-inspired training loss function.

Information-theory based forest training function. The most common approach for training split nodes (see Figure 7.1) in classification trees tries to select the values of parameters ϕ_j in each internal node j by *maximizing* an information gain criterion G, defined as

$$G(\mathcal{Z}_j, \phi_j) = H(\mathcal{Z}_j) - \sum_{i \in \{\mathtt{l},\mathtt{r}\}} \frac{|\mathcal{Z}_j^i|}{|\mathcal{Z}_j|} H(\mathcal{Z}_j^i)\,, \tag{7.7}$$

with the entropy being defined as $H(\mathcal{Z}) = -\sum_{c \in \mathcal{Y}} p(c|\mathcal{Z}) \log p(c|\mathcal{Z})$. The first term in (7.7) is constant with respect to the parameters ϕ.

From here we can see that the parameter learning concept, when driven by information gain maximization, is intimately related to a more general *energy minimization* problem, formulated as:

$$\phi_j = \arg \min_{\phi' \in \Psi(\mathcal{Z})} E(\mathcal{Z}_j, \phi')\,, \tag{7.8}$$

where $\Psi(\mathcal{Z})$ is the set of randomly generated split parameters to be tested and $E(\cdot)$ is an energy function to be defined.

When we now drop the first term and also the normalization constant in the second term of (7.7), we may re-write it in terms of an *information-theoretic* energy

$$\begin{aligned} E_{\mathtt{IT}}(\mathcal{Z}_j, \phi_j) &= \sum_{i \in \{\mathtt{l},\mathtt{r}\}} |\mathcal{Z}_j^i| H(\mathcal{Z}_j^i) \\ &= \sum_{i \in \{\mathtt{l},\mathtt{r}\}} -|\mathcal{Z}_j^i| \sum_{c \in \mathcal{Y}} p(c|\mathcal{Z}_j^i) \log p(c|\mathcal{Z}_j^i) \\ &= -\sum_{i \in \{\mathtt{l},\mathtt{r}\}} \sum_{c \in \mathcal{Y}} n(c, \mathcal{Z}_j^i) \log \frac{n(c, \mathcal{Z}_j^i)}{|\mathcal{Z}_j^i|} \end{aligned} \tag{7.9}$$

where

$$n(c, \mathcal{Z}) = \sum_{(x_i, c_i) \in \mathcal{Z}} \mathbb{1}\left[c_i = c\right] \tag{7.10}$$

denotes the number of training samples of class c in $\mathcal{Z}$ and $p(c|\mathcal{Z}) = \frac{n(c,\mathcal{Z})}{|\mathcal{Z}|}$ is the empirically estimated probability for class c. This can be obtained by e.g. using a histogram representation.

Field-inspired forest training objective. Alternatively, we can think of training a split node j in the trees by using an MRF-like energy $E_{\texttt{RF}}$, which we define as

$$E_{\texttt{RF}}(\mathcal{Z}_j, \phi_j) = \sum_{i \in \{\texttt{l},\texttt{r}\}} \left(\sum_{(x_k,c_k) \in \mathcal{Z}_j^i} \psi(c_k; \mathcal{Z}_j^i) + \lambda \sum_{z_k=(\mathbf{u}_k,c_k)\in\mathcal{Z}_j^i, \mathbf{q}\in\mathcal{N}(\mathbf{u}_k)} \psi'(z_k, \mathbf{q}) \right) \tag{7.11}$$

with $\mathcal{N}(\mathbf{u}_k)$ denoting a local neighborhood of the point $\mathbf{u}_k$. As unary potentials we choose the commonly used log-loss

$$\psi(y; \mathcal{Z}) = -\sum_{c\in\mathcal{Y}} \mathbb{1}\left[y = c\right] \log p(y|\mathcal{Z})\,. \tag{7.12}$$

If we ignore the pairwise term (by setting $\lambda = 0$) we get

$$\begin{aligned} E_{\texttt{RF}}(\mathcal{Z}_j, \phi_j) &= -\sum_{i\in\{\texttt{l},\texttt{r}\}} \sum_{(x_k,c_k)\in\mathcal{Z}_j^i} \sum_{c\in\mathcal{Y}} \mathbb{1}\left[c_k = c\right] \log p(c_k|\mathcal{Z}_j^i) \\ &= -\sum_{i\in\{\texttt{l},\texttt{r}\}} \Bigg(n(c=1, \mathcal{Z}_j^i) \log \frac{n(c=1,\mathcal{Z}_j^i)}{|\mathcal{Z}_j^i|} + \ldots \\ &\quad + n(c=C, \mathcal{Z}_j^i) \log \frac{n(c=C,\mathcal{Z}_j^i)}{|\mathcal{Z}_j^i|} \Bigg) \\ &= -\sum_{i\in\{\texttt{l},\texttt{r}\}} \sum_{c\in\mathcal{Y}} n(c, \mathcal{Z}_j^i) \log \frac{n(c,\mathcal{Z}_j^i)}{|\mathcal{S}_j^i|}\,. \end{aligned} \tag{7.13}$$

We discover that under the above assumptions (7.13) and (7.9) are identical. Thus, conventional entropy-based tree training corresponds *exactly* to minimizing a field-like energy which uses the log-loss as unary and no pairwise term. However, the more interesting findings come when we consider the effect of having unbalanced classes in the training set, which is discussed next.

7.5.2 Correcting class imbalance

For the case of unbalanced classes in the training data (the training samples are not homogeneously distributed across categories), introducing a global correction term was shown to be effective, especially for the task of semantic image segmentation [Shotton et al., 2008a]. One of the reasons is that e.g. the pixels belonging to the background (or *stuff*) class(es) are much more numerous compared to "object" (or *thing*) class(es). To this end, let us consider the general case of a non-balanced training set $\mathcal{Z}_0$, (initially) available at a root node of the tree construction process where $n(c_i, \mathcal{Z}_0) \neq n(c_j, \mathcal{Z}_0),\ c_i, c_j \in \mathcal{Y}, i \neq j$.

Similar to the previous section, we will analyze the respective objective functions (information theoretic and random field-inspired), but now using the following global, per-class re-balancing weights

$$\omega_c = \frac{\sum_{k \in \mathcal{Y}} n(k, \mathcal{Z}_0)}{n(c, \mathcal{Z}_0)} = \frac{|\mathcal{Z}_0|}{n(c, \mathcal{Z}_0)}\,, \tag{7.14}$$

where $\mathcal{Z}_0$ is the whole training data used to grow a tree, i.e. the data available to the root node. Moreover, we will use the following, node-based normalization factor

$$W(\mathcal{Z}_j) = \sum_{k \in \mathcal{Y}} \omega_k \; n(k, \mathcal{Z}_j)\,. \tag{7.15}$$

Information-theory based forest training function. When applying the reweighing terms and performing some algebraic manipulation, the information-theoretic objective straightforwardly changes to

$$\begin{aligned} E_{\mathtt{IT}}(\mathcal{Z}_j, \phi_j) &= \sum_{i \in \{\mathtt{l},\mathtt{r}\}} -W(\mathcal{Z}_j^i) \sum_{c \in \mathcal{Y}} \frac{w_c \, n(c, \mathcal{Z}_j^i)}{W(\mathcal{Z}_j^i)} \log \frac{w_c \, n(c, \mathcal{Z}_j^i)}{W(\mathcal{Z}_j^i)} \\ &= -\sum_{i \in \{\mathtt{l},\mathtt{r}\}} \sum_{c \in \mathcal{Y}} w_c \, n(c, \mathcal{Z}_j^i) \; \log \frac{w_c \, n(c, \mathcal{Z}_j^i)}{W(\mathcal{Z}_j^i)}\,. \end{aligned} \tag{7.16}$$

Field-inspired forest training objective. In contrast, for the proposed, field-inspired training objective, class reweighing results in the following formulation:

$$\begin{aligned} E_{\mathtt{RF}}(\mathcal{Z}_j, \phi_j) &= -\sum_{i \in \{\mathtt{l},\mathtt{r}\}} W(\mathcal{Z}_j^i) \sum_{(x_k, c_k) \in \mathcal{Z}_j^i} \sum_{c \in \mathcal{Y}} \mathbb{1}\,[c_k = c] \log p(c_k | \mathcal{Z}_j^i, \omega) \\ &= -\sum_{i \in \{\mathtt{l},\mathtt{r}\}} W(\mathcal{Z}_j^i) \left(n(c=1, \mathcal{Z}_j^i) \log \frac{\omega_{c=1} n(c=1, \mathcal{Z}_j^i)}{W(\mathcal{Z}_j^i)} + \ldots \right. \\ &\qquad \left. + \, n(c=C, \mathcal{Z}_j^i) \log \frac{\omega_{c=C} n(c=C, \mathcal{Z}_j^i)}{W(\mathcal{Z}_j^i)} \right) \\ &= -\sum_{i \in \{\mathtt{l},\mathtt{r}\}} W(\mathcal{Z}_j^i) \sum_{c \in \mathcal{Y}} n(c, \mathcal{Z}_j^i) \; \log \frac{w_c \, n(c, \mathcal{Z}_j^i)}{W(\mathcal{Z}_j^i)}\,. \end{aligned} \tag{7.17}$$

Thus, after class re-balancing, the entropy-based energy in (7.16) and the field unary in (7.17) are no longer the same. Interestingly, the per-class, global reweighing term ω_c, which induces the global importance weight for each sample in (7.16) vanishes from the inner sum in the random field inspired version (7.17), while it still influences the way the probability mass function is estimated. Consequently, we can observe that the node split parameter optimization process is more directly guided by the factual number of samples reaching a particular node, even though their global weights could suggest to diminish their overall contribution. In this sense, we can notice an important transition in the paradigm of the training process: The conventional, information theoretic approach propagates the global reweighing terms through the entire tree/forest growing process while the introduced random-field inspired term puts stronger emphasis on the composition of the training data, reaching the respective nodes.

In the next section we show the effect on accuracy of both: i) using entangled geodesic features, and ii) training the forest by minimizing (7.17) rather than (7.16).

7.6 Results and Comparisons

This section assesses the accuracy and versatility of our segmentation approach. To this end, we evaluate on the CamVid dataset as introduced in the previous chapter of this Thesis and on three novel, labelled image datasets which we introduce and discuss in the next sections, respectively.

7.6.1 Image datasets

An overview on the versatility of the data we have used in our experiments is shown in Figure 7.5. Now, we give a short description on each dataset. **LFW: Labelled Faces in the Wild**. This is an augmented version of the public dataset in [LFW, 2013], where we have manually segmented a subset of 1250 images into the following 8 categories: background, nose, mouth, L/R eye, L/R eyebrow and lower face. The contained faces exhibit strong variations in pose and appearance. Furthermore, the mouth and eyes show considerable articulation.

CT: Computed Tomography. We tested our algorithm also against a new dataset of medical images. It comprises of 2D coronal slices taken at random positions of labelled, 3D CT scans. As ground truth, different anatomical entities have been segmented in 3D, using an interactive segmentation tool. We have the following 9 classes: background (BG), heart (HR), liver (LI), spleen (SP), left/right lung (LL/RL), left/right kidney (LK/RK) and aorta (AO).

KinBG depth images. This is a newly created dataset, similar to the body-part Kinect training dataset used in [Shotton et al., 2012], with the difference that the retargeted MoCap characters have been inserted within a Kinect acquired, real background scene. We are using ground truth labels for 12 body parts (L/R head side, neck, torso, L/R arm, L/R hand, L/R leg, L/R foot) and three background classes. In fact, in contrast to [Shotton et al., 2012], we do not assume a given FG/BG separation, and the background is subdivided into: floor, back wall and general background. This yields a total of 15 categories.

7.6.2 Comparisons with related methods

We provide comparisons with various state-of-the-art decision forest based approaches [Montillo et al., 2011, Kontschieder et al., 2011, Yang et al., 2012]. We

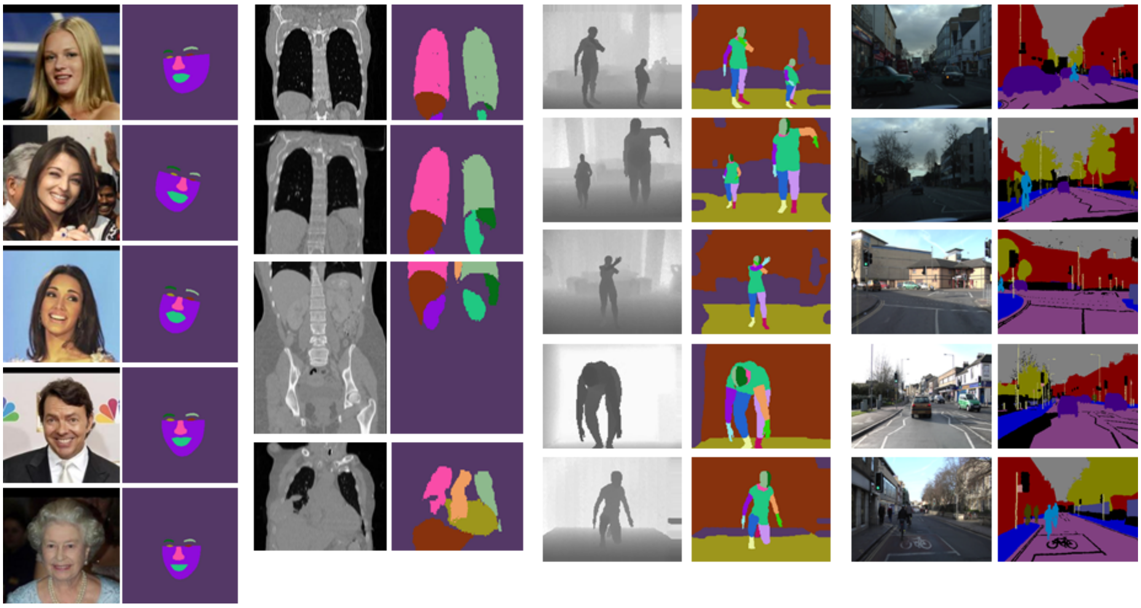


Figure 7.5: Illustration of datasets where we have evaluated our proposed approach. From left to right: Labelled Faces in the Wild, Computed Tomography, KinBG depth images and CamVid road scene datasets with their corresponding ground truth annotations.

also compare against approaches using forest-based unaries followed by CRF smoothing [Kolmogorov, 2006]. In the latter, as energy model, we used a log-loss in the unary term and a contrast-sensitive Potts model in the pairwise term. Additionally, we also implemented an auto-context [Tu and Bai, 2010] version of a random forest: A first classification forest is trained using raw intensity features. A second forest is then trained using as features both intensities and the class posteriors obtained from the first forest. In order to ensure a fair comparison, we train all forest-based algorithms to the same number of nodes. The parameters of all baseline algorithms have been optimized so as to yield the highest Jaccard scores.

Quantitative results are summarized in Table 7.6.2 where we compare algorithms both in terms of accuracy and runtime. Segmentation accuracy is computed via the Jaccard score (as adopted also in the PASCAL VOC challenge [VOC, 2013]). Runtimes are reported for similarly non-optimized C# implementations. However, decision forests are well-suited for GPU implementations [Sharp, 2008]. For all forest based algorithms we fix $T = 10$ and $D = 20$, except for the CamVid dataset where we use a maximum depth $D = 17$ since the number of training samples is considerably smaller.

Labelled Faces in the Wild. The baseline forest 01 yields a mean Jaccard score of only 38.1% as it produces noisy segmentations and overly bold segments for the smaller objects such as the eyebrows (cf. Figure 7.6 (c)). CRF-based post-processing 02 boosts the score to 45.2%, still lower than what our implemented auto-context forest 03 and

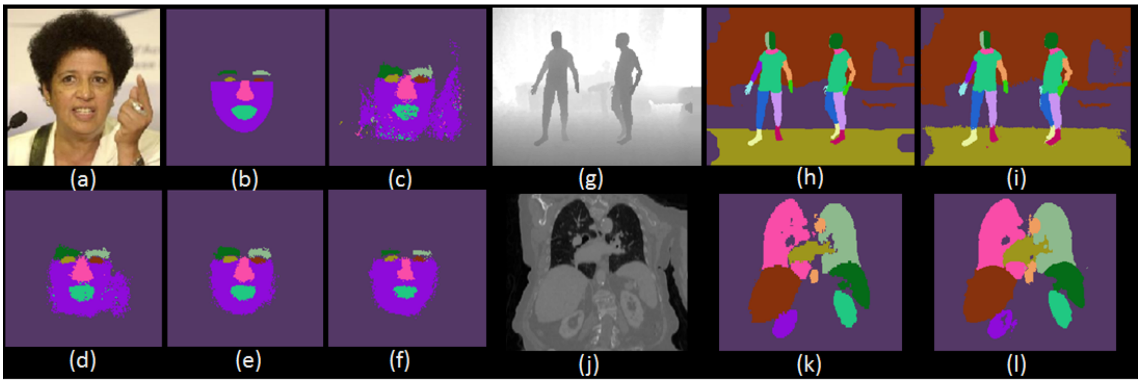


Figure 7.6: **The effect of geodesic entanglement on spatial coherence. (a, g, j)** Input test images, from the LFW, KinBG and CT datasets, respectively. **(b, h, k)** Ground truth labels (different colors for different classes). **(c)** Segmentation results from conventional pixel-wise classification forest. The lack of spatial smoothing produces noisy labeling. Notice also the overly large eye/eyebrow segments. **(d)** Results from forest with probability entanglement. Entangling the p channels only helps spatial coherence of the output. **(e)** Results from forest with geodesic entanglement. Enabling the long-range geodesic feature channels g helps spatial coherence further. The spurious hand region is gone. **(f, i, l)** Results from forest with geodesic entanglement and field-inspired energy term. Using our field-inspired energy term helps further still. e.g. notice the better recovered eye shape in (f).

our proposed geodesic forests achieve ((07)-(16)). Both the use of geodesic features and the field-inspired energy help achieve the highest accuracy in this dataset. As shown in Figure 7.6 (f), our geodesic forests better delineate small structures.

Figure 7.7 plots the (testing) accuracy of algorithms ((01), (08), (14) and (16)) as a function of the tree depth. Entangled geodesic forests using either of the two energy models ((14), (16)) work better than the conventional forest (01). Using the field-inspired energy (16) works better than the conventional information gain (14). Using two entanglement sections works better than a single one on this data. Our auto-context geodesic forest (08) does well, but interestingly the presence of a second forest does not seem to improve things much compared to using one forest only.

In terms of runtime, the standard forest + CRF (02) takes ≈ 0.71s (per frame) versus ≈ 0.42s for a single-section entangled geodesic forest. Also, forest-based inference is simpler and more easily parallelizable than using graph-cut algorithms for inference on CRF.

CT scans. Starting with baseline scores of 53.2% (01) and 68.3% (02) we find again that providing geodesic features improves on all our compared methods. The auto-context forest performs well too, even without these additional features. However, the best results are achieved with one or two sections of entanglement in geodesic forests ((12), (16)). The

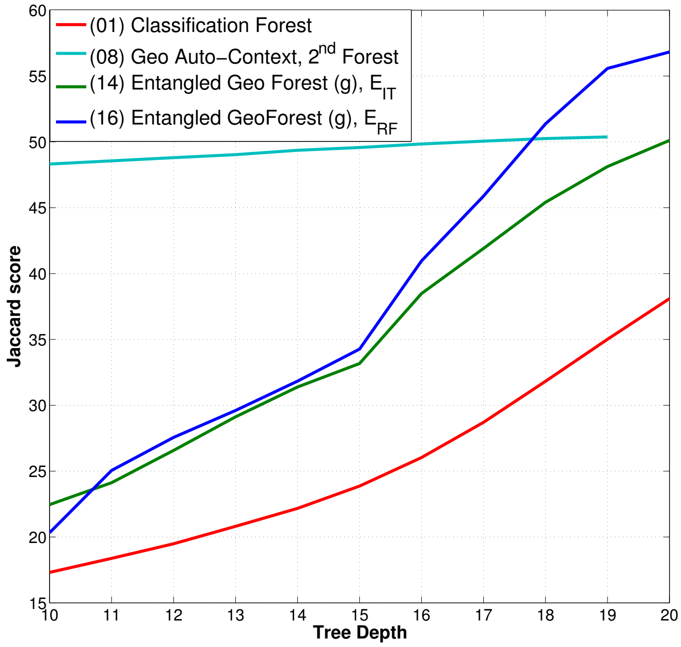


Figure 7.7: **Accuracy as a function of tree depth** D, for different forest variants, evaluated on LFW dataset.

CRF approach 02 takes ≈ 1.2s per frame while geodesic forests 12 needs ≈ 0.72s.

KinBG depth images. In this dataset the best results are achieved by our auto-context, geodesic forests (07,08) which provide strong improvements over the baseline (+6.8% over 01, + 3.9% over 02). However, note that using auto-context forest variants (e.g. 03, 07, 08) results in higher runtimes as two forests need to be evaluated (resulting in ≈ 1.39s/frame). The CRF approach 02 takes ≈ 1.35s per frame while entangled geodesic forests are much faster (≈ 0.64s/frame). In contrast to [Shotton et al., 2012], we achieve body part *and* background class labeling without the need for a preliminary background separation stage.

CamVid videos. For this dataset we have followed the experimental setup described in [Kontschieder et al., 2011], providing Lab raw channel intensities, first and second order image gradients and HOG-like features. The baseline result for 01 is 33.3% which we are able to considerably outperform with all our geodesic forest variants. The best performing geodesic forest 16 improves over the recent work in [Kontschieder et al., 2011] (+2.1%) and [Yang et al., 2012] (+8.7%). The highest score is obtained by the CRF 02 (41.7%), but at the expense of twice the runtime: ≈ 1.07s/frame for 02 and ≈ 0.56s/frame for geodesic forests.

Smoother energy models? In further experiments we have also tried training forests

	Image datasets			
	LFW	**CT**	**KinBG**	**CamVid**
Number of training / testing images	1000 / 250	512 / 250	2500 / 250	367 / 233
Accuracy of existing algorithms				
01 Classification forest (pixel-wise classification only, no spatial context, no geodesic features)	38.1	53.2	57.1	33.3
02 Classification forest + conditional random field (CRF)	45.2	68.3	60.0	**41.7**
03 Auto-context classification forest (energy E_{IT}; no geo. features)	48.1	65.9	61.9	35.2
04 Entangled classification forest (energy E_{IT}; no geo. features; single entgl. section depth=10)	43.2	58.3	55.7	35.5
05 Structured class-labels in random forests [Kontschieder et al., 2011]	–	–	–	36.2
06 Local label descriptor [Yang et al., 2012]	–	–	–	29.6
Accuracy of variants of proposed geodesic forests (GeoF)				
(*Two autocontext forests*)				
07 Auto-context geodesic forests (energy: E_{IT}; evaluation using raw class posterior p as output)	49.8	65.7	**62.4**	35.2
08 Auto-context geodesic forests (energy: E_{IT}; evaluation using smooth class posterior g as output)	50.4	69.2	**63.9**	36.6
(*One entanglement section at depth=10.*)				
09 Entangled geodesic forests (energy E_{IT}; evaluat. p)	46.8	58.6	55.9	36.8
10 Entangled geodesic forests (energy E_{IT}; evaluat. g)	46.2	60.2	55.4	35.1
11 Entangled geodesic forests (energy E_{RF}; evaluat. p)	54.3	69.1	59.8	34.9
12 Entangled geodesic forests (energy E_{RF}; evaluat. g)	54.6	**72.3**	60.0	37.7
(*Two entanglement sections, at depth=10 and depth=15.*)				
13 Entangled geodesic forests (energy E_{IT}; evaluat. p)	49.5	60.3	56.6	37.9
14 Entangled geodesic forests (energy E_{IT}; evaluat. g)	50.1	61.1	56.8	38.0
15 Entangled geodesic forests (energy E_{RF}; evaluat. p)	**56.6**	69.9	59.8	36.6
16 Entangled geodesic forests (energy E_{RF}; evaluat. g)	**56.8**	**72.2**	60.3	**38.3**

Table 7.1: **Quantitative validation and comparison.** Average Jaccard accuracy measures (in %, larger values are better) across all classes, for our geodesic forest algorithm as compared to existing techniques (e.g. random classification forest, and forest + CRF), for four different labelled image databases. Bold-face numbers indicate the top two algorithms for each image dataset.

by adding pairwise terms or other global smoothness terms in the energy (7.17), but without being able to consistently improve the accuracy further. These results suggest that our long-range connectivity features may already do a good job at imposing spatial smoothness.

Capturing semantic context via entangled geodesic features. Figure 7.8 illustrates how geodesic forests capture long-range semantic context on the computed tomography dataset. For a reference pixel of a given class (e.g. liver) the elements of each matrix indicate the frequency of classes in the two automatically selected probes (e.g. probe 1 in the rows and probe 2 in the columns). For example, in Figure 7.8a we see that at depth 10 (after first level of entanglement) when the reference pixel is in the liver, the two probes tend to be selected (during training) to also be in the liver. This encourages local context and label smoothing; and can be thought of a generalization of MRF where the discriminative cliques are learned rather than being predefined. For deeper trees we start to see the effect of longer-range semantic context. In fact, e.g. in Figure 7.8b we observe that the probes tend to be selected frequently also in the heart and right lung regions. This indeed makes sense when the goal is to identify liver pixels. Similar reasoning applies to other classes (e.g. see Figure 7.8a',b',c' for pixels in the l. kidney).

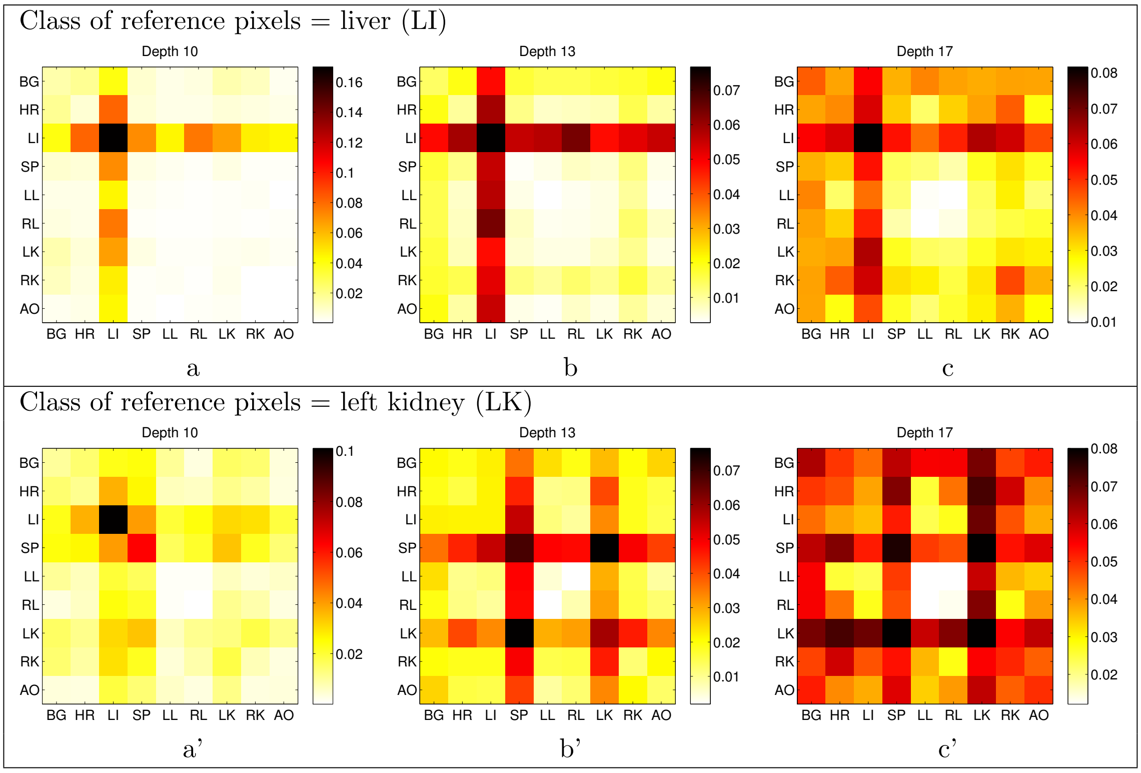


Figure 7.8: **Class co-occurrence matrices for the two feature probes (a,b,c)** For a reference point is in the liver. **(a',b',c')** For a reference point is in the left kidney. Co-occurrence matrices are shown for three different tree depths: $D = 10$, $D = 13$, $D = 17$. In this dataset (CT) classes are: background (BG), heart (HR), liver (LI), spleen (SP), l./r. lung(LL/RL), l./r. kidney (LK/RK) and aorta (AO). This figure demonstrates capturing semantic context. e.g. in b' when trying to identify the left kidney it helps to use probes either in the spleen region (just above the left kidney) or in the left kidney itself (encouraging local spatial smoothness).

7.6.3 More Qualitative Segmentation Results

Here, we provide further segmentation results on test images from the four datasets used in this work, obtained by the introduced combination of entangled geodesic features and the new training objective function used in the tree growing process. The results are provided in tabular form where each row represents a different test image, arranged as original image, ground truth and output probabilities $g(c|I)$ as a function of the maximal tree depth.

Labelled Faces in the Wild (LFW) (Table 7.6.3) We see that even shallow trees ($D = 10, 12, 14$) capture the rough positions of the various facial features quite well. However, as the depth increases we get both, more precise feature segmentation *and* better handling of e.g. rotation of the head (see first two rows).

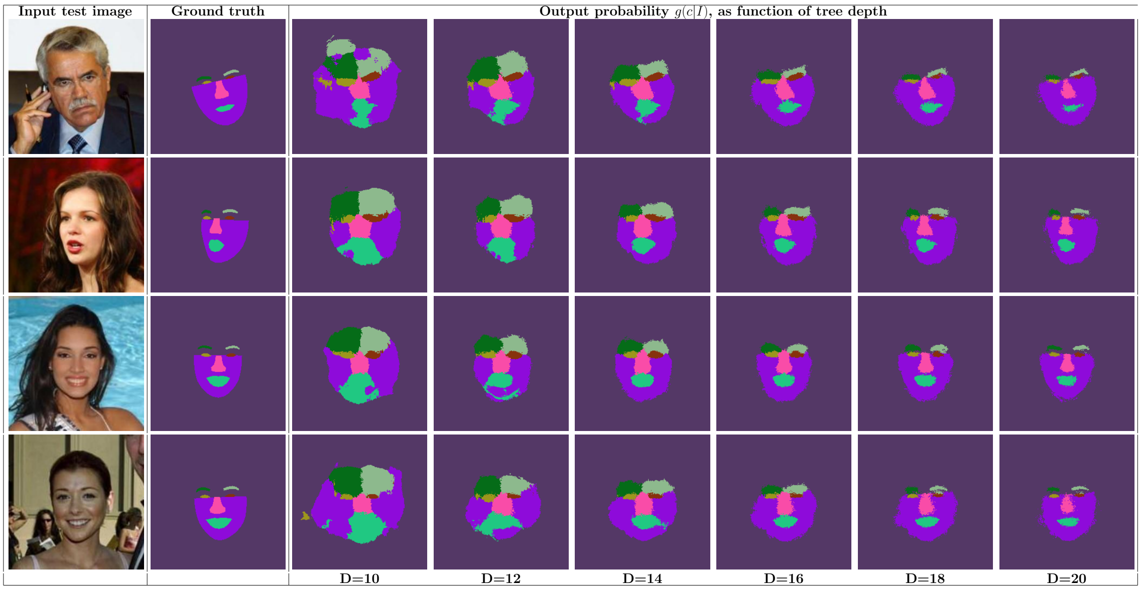


Table 7.2: **Segmentation results obtained by GeoF on the faces dataset.** See text for details.

Computed Tomography Dataset (CT) (Table 7.6.3) As the forest depth increases we get ever more accurate delineations of the internal organs such as the lungs, heart and kidneys. Note that accurate and smooth segmentation is achieved also for the challenging, thin and tubular-shaped aorta (orange label). In this sense, we demonstrate the general applicability of our method and that there are no implicit assumptions about shape or composition of objects to be segmented.

KinBG depth images Dataset (Table 7.6.3) This dataset is different from the one used in [Shotton et al., 2011] in a sense that multiple, computer generated characters have been realistically inserted within depth images of background scenes (containing floor, sofas, curtains etc.). The background itself has been partitioned into three ground-truth classes: floor, back wall and everything else. In all cases **GeoF** produces convincing segmentation results where all background and foreground classes are segmented automatically and simultaneously. Notice how deeper trees produces more accurate delineation and recognition of the smaller body parts such as hands and feet, and reduce "bleeding" into the floor category.

CamVid Dataset (Table 7.6.3) Finally, we provide some qualitative results obtained on the driving videos (CamVid) dataset. Once again, deeper forests produce increased accuracy, especially for the smaller and more challenging objects.

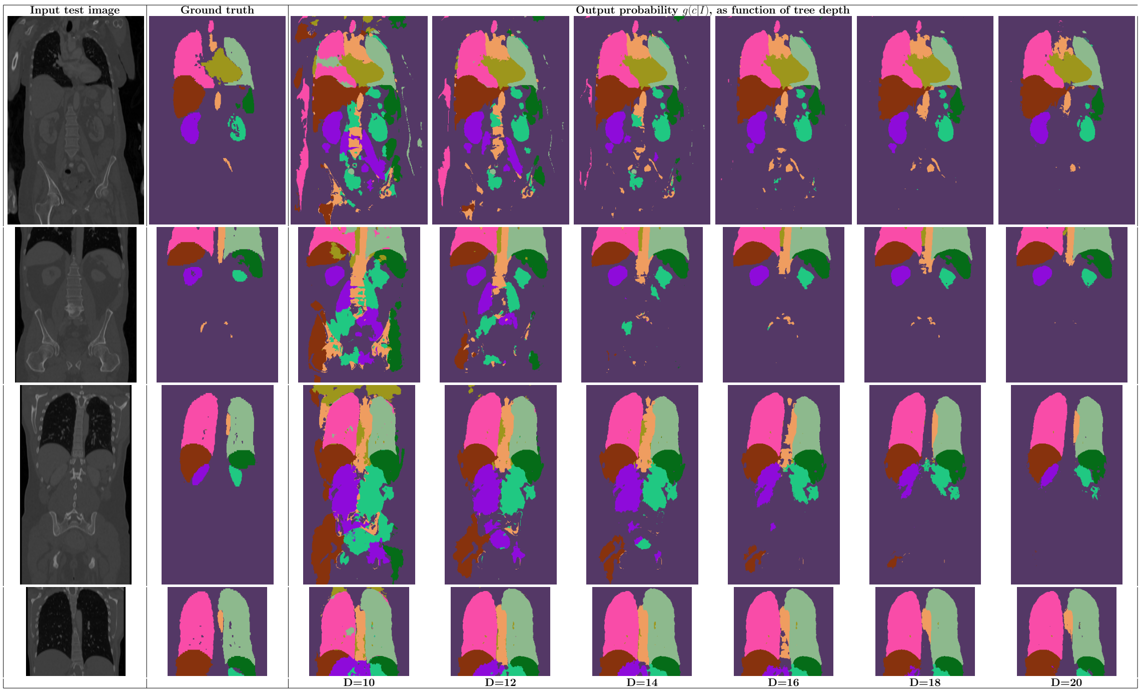


Table 7.3: **Segmentation results obtained by GeoF on the computed tomography dataset.** See text for details.

7.7 Conclusion

This chapter has presented a new forest-based model for structured learning, applied to the task of semantic image segmentation. Our model encourages spatial smoothness and long-range, semantic context within the forest itself, via the use of new, soft connectivity features which build upon entangled, generalized geodesic distances. In addition, we showed how training forests by minimizing a random field-inspired energy yields higher accuracy than information gain based approaches. Quantitative validation on four diverse image datasets shows at par or better accuracy than state-of-the-art approaches including pairwise conditional random fields, with faster runtimes.

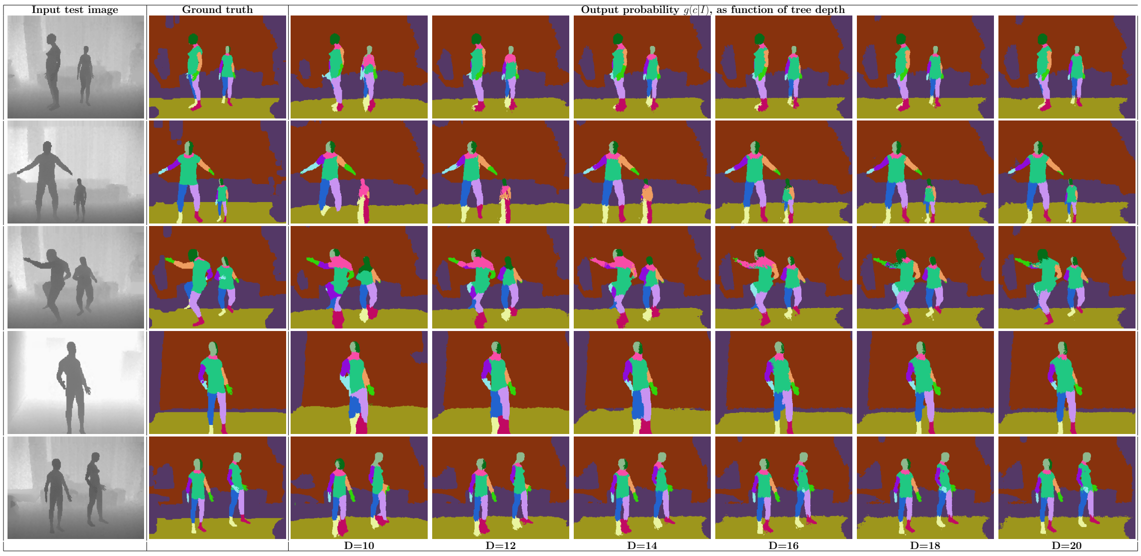


Table 7.4: **Segmentation results obtained by GeoF on the background-augmented depth image dataset.** See text for details.

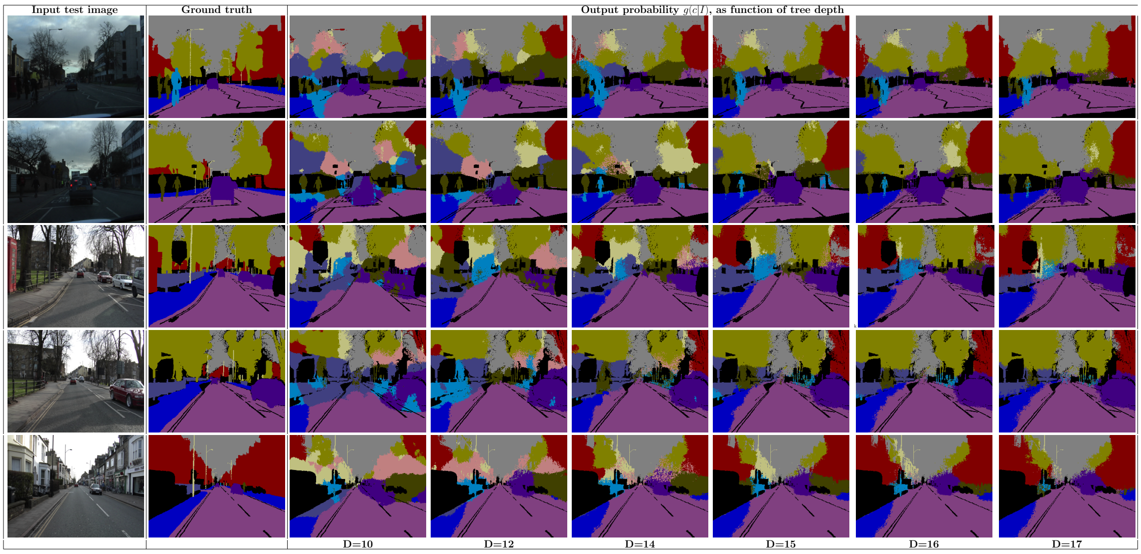


Table 7.5: **Segmentation results obtained by GeoF on the driving videos (CamVid) dataset.** See text for details.

Chapter 8

Conclusion

Contents

8.1 Summary

In this Thesis we have undertaken a modern view on random decision forests. Decision forests are ensembles of binary decision trees and provide a general framework for machine learning problems. Here, we have dealt with their application for offline, supervised learning tasks, i.e. at the beginning of the training process all data is available together with ground truth information. More specifically, we investigated their suitability for two fundamental computer vision problems, object detection and semantic image segmentation.

When applying random forests for these particular visual computing tasks, the underlying problems are typically reduced to *classification* or mixed *classification and regression* problems for semantic segmentation and object detection, respectively. This is mainly due to the fact that random forests are conveniently applicable and available as off-the-shelf learners for such problems. Moreover, they were empirically shown to perform favourable when compared to machine learning techniques like support vector machines (SVM), k-nearest neighbors (k-NN) or boosting, especially on high-dimensional data. Other interesting properties are their inherent ability to handle multi-label problems, little tendency to overfitting while it is possible to parallelize training and evaluation procedures.

Although forests already hold many desired properties, we have investigated the question on whether it is possible to improve their performance on problems where the data to be processed is structured, as it is often the case in the visual computing domain. With structured, we mean data where spatially adjacent samples typically exhibit a high degree of correlation. Since image data (like plain 2D-images, image sequences, video streams, medical image data sets, etc.) is mostly organized on a regular pixel/voxel grid, we can speak of directed adjacency, respecting the ordering of data samples. In fact, we recognize that also certain semantic and dependency properties are ordered and therefore structured in this data. From a more general point of view, one might declare this as *contextual information* which we propose to be subsequently exploited when using random forests in computer vision.

The main goal of this Thesis was to render random forests context-aware, in order to more efficiently exploit the previously described structure in the data. However, before our contributions are described in the two main parts of the Thesis, we introduced the general theory behind them in Chapter 2 and showed specific instantiations to perform classification, regression and mixed classification and regression. Classification forests have a discrete output space, yielding to categorical decisions while regression forests typically predict continuous-valued outputs. Another briefly introduced variant of random forests are *random ferns*, which are a non-hierarchical version in the sense that all binary tests applied at the same level are identical.

8.1.1 Summary on Object Detection

The first main part of this Thesis is dedicated to the object detection problem, starting with Chapter 3. There, we investigated the task of shape-based object detection and discussed the problem of shape data representation when learned in random forests. To this end, we introduced a novel shape fragment descriptor that abstracts and describes local shape information by means of angular relations between connected edge points. We have evaluated several properties of the descriptor including distinctiveness, efficacy, invariance, efficiency and demonstrated its suitability to be learned in random forests together with their relative offset position to the object centroid.

In the next Chapter (4), we have discussed the task of non-maximum suppression (NMS), i.e. a post-processing step often necessary for predictor outputs using the concept of generalized Hough-voting, such as Hough forests. We have introduced *Evolutionary Hough Games* which describe an evolutionary game-theoretic approach for detecting (multiple) instances of an object category by the set of voting elements (pixels, patches, shape fragments, etc.) they are composed by. The approach is twofold: First, we train a standard Hough forest that stores object class distributions and offset voting information in the leaf nodes of the trees. At test time, we construct a payoff matrix holding pairwise compatibilities of test samples, obtained by analyzing their mutual geometric compatibilities with respect to the learned object category model. To this end, we exploit the available leaf statistics and relate them with the respective test sample positions in the image under test. Once the payoff matrix is constructed, we run an evolutionary game theoretic dynamics to identify the set of mutually best compatible elements, forming an evolutionary stable set of strategies (in the terminology of game theory). To obtain multiple object hypotheses, we proposed a novel enumeration scheme, inspired by *tabu search*. The basic idea is to eliminate already found solutions from the search space when subsequent objects shall be identified. However, instead of truncating or altering the payoff matrix (which would result in a change of the game), we initialize subsequent search attempts on the original matrix with solutions found on the restricted game. As a result, we provide detections that are not defined by their enclosing bounding boxes but instead directly identify their contributing samples from contextual relations.

While in the first two chapters of the detection part, the random forest model was considered as a black-box learning algorithm, we introduced *Context-sensitive decision forests* in Chapter 5, which are a novel way to integrate contextual information into the training and evaluation processes. We departed from the traditional approach where trees are used as simple predictors and instead allowed them to access intermediate predictions for the data of themselves during training and testing. From a methodological point of view, this concept is similar to famous models like Auto-Context or feed-forward networks, however, we have demonstrated that it can be efficiently encapsulated in the trees without the need of several predictor stages. Moreover, we have introduced a novel loss function which effectively handles classification and regression tasks in a joint formulation. Based on the loss that is constantly monitored for the non-leaf nodes, we introduced *prioritized tree growing*, i.e. a mechanism that continues tree growing where the error with respect to the ground truth is currently the highest. In addition, the way we split data that is equipped with regression information (we used object centroid voting vectors for foreground samples) follows a mode isolation technique in order to better separate multi modal data. Finally, we introduced a new family of split functions which is able to exploit training data containing several, possibly mutually occluding object instances. In such a way, we provided a context-driven way of learning random decision trees, i.e. better context from isolated modes obtained by joint loss consideration lead to better predictors.

8.1.2 Summary on Semantic Image Segmentation

The second main part of this Thesis is dedicated to the semantic image segmentation problem. In semantic image segmentation, the task is to correctly assign a categorical class label to each pixel in an image under test and therefore obtain coherently segmented regions and objects. Starting in Chapter 6, we introduced a way to include local context in the random forest training process by exploiting so-called structured class-labels, obtained from ground truth information. Traditionally, each training sample of a random decision tree is associated with a single, *atomic* class label and also the prediction step classifies a single pixel. However, since for the semantic segmentation task the output domain is obviously structured in the sense that neighboring pixels often have the same labels, we proposed a variant that also takes structured class-labels into account for both, training and testing. When learning the parameters in the split nodes, we evaluated joint statistics of k-label joint distributions of randomly selected label positions within the structured class-labels. As a result, the trees were aiming to cluster together label patches, that allowed to i) predict labels also for the neighbors covered by the label patches and ii) capture valid label transitions of the ground truth in order to prevent from propagating semantically implausible labellings. We have introduced two ways for integrating structured label predictions into concise per-pixel labellings. The first way decides for the final label with a simple majority voting step that integrates predictions from neighborhoods covered by the label patches. The second approach is based on this simple fusion: After generating a new per-pixel labelling, we allow an alternative selection among the predictions provided by the forest ensemble. Afterwards, the new labelling is again produced by another simple fusion process. In this sense, this process is alternated until convergence.

With the approach presented in Chapter 6, we could show improved segmentation results in terms of accuracy and introduced a mechanism for predicting structured labels in random forests. However, the inference process requires additional computation for producing the final labelling, therefore cutting back on the evaluation speed properties of the trees. In Chapter 7 we investigate the semantic image segmentation problem from yet another perspective within random forests, maintaining the original computational complexity during test time: Their main reason for computational efficiency is the individual (and therefore possibly parallel) treatment of all pixels. However, this approach prevents them from interrelating per-pixel outputs although they are clearly linked to each other when considering the segmentation task. This shortcoming is typically compensated by employing random field like models as post-processors, which establish neighborhood relations in terms of (mostly) parametric potential functions. In this chapter we have introduced a novel method to encourage spatial consistency in the obtained labellings. To this end, we revealed hidden correlations in the output space by encoding long-range, soft-connectivity features between pixels in the feature space the trees learn from. These features can be efficiently computed by using generalized geodesic distance transform, i.e. the computation can be approximated by simple, linear forward- and backward-kernel operations on the generated classifier outputs. Moreover, these geodesically smoothed predictor outputs are contrast-sensitive in terms of test image gradients and therefore mimic a random field-like regularization as performed in conditional random fields. Feeding back intermediate predictor outputs in the learning process in this way is another contribution of this chapter as it allows the learner to compensate mis-labellings in the subsequent levels of the trees. The second major contribution of this chapter is in the way the split functions are designed. To enforce spatial consistency also in the way the trees are constructed, we have established a rigorous connection between traditionally used, information-theoretic objective functions and a simple Markov random field (MRF) energy. We have shown their equality when choosing a particular energy type in the MRF, i.e. the prominent *negative log-loss* in the unary term and no pairwise term. Finally, we presented how the objective function is altered when non-balanced training data is processed, yielding to a significant improvement in the output quality of the predictions.

8.2 Conclusions and Outlook for Future Work

One of the most important take home messages of this Thesis is that machine learning on computer vision data is supposed to exploit the data structure, inherently available to it. This is in contrast to the widespread notion of considering learning algorithms as simple black-box toolboxes. We have demonstrated that the choice of an effective learning method like the random decision forest allows to efficiently integrate knowledge about this structure or *contextual information*, without substantially increasing the computational complexity. Furthermore, we have provided empirical evidence in terms of improved recognition and segmentation scores on many and diverse standard benchmark datasets, when integrating contextual cues within the random forest learner.

In our evaluations, we have hardly experienced limitations of the general learning concept: While it has been shown that random forests scale well with an increasing amount of data [Shotton et al., 2012], there is yet no study reporting performance on prediction tasks with very high-dimensional output spaces. For example, an interesting direction might be to see how learning of extremely large scale datasets like the ImageNet dataset [Ima, 2012] can be performed, where the goal is to classify and label subsets of the 10.000.000 image-database into 10.000+ object categories. Currently top-ranked methods like [Krizhevsky et al., 2012] employ deep learning strategies which also integrate contextual cues over multiple layers and obtain excellent results.

Another potential future research direction could address the topic of choosing proper loss functions in the split nodes and how to appropriately optimize them. While standard entropy measures seem to do a reasonable job for non-structured data, specific computer vision tasks demand customized solutions where the problem nature can directly guide the objective function formulation, as we have shown e.g. in Chapters 5 and 7. In this line, recent work [Schulter et al., 2013] focuses on the formulation of global loss functions via maintaining an adaptive weight distribution over the training samples during the training process. More complex loss functions have not yet been successfully reported for decision trees, however, the work of [Tarlow and Zemel, 2012] introduces some interesting concepts to be used within structured SVM. The problem of non-greedy node split optimization is (to some extend [Nowozin et al., 2011, Montillo et al., 2013]) largely unexplored and defines another interesting direction for future works. However, one of the major problems to be solved here is the compromise between maintaining generalization properties *and* avoid overfitting to the data.

Finally, proponents of random forests are often confronted with the lack of a clean and crisp theory as e.g. available for SVM. This lack is mainly arising due to their non-deterministic nature, however, [Breiman, 1996b] introduced a mechanism to determine the so-called *Out-of-Bag-Error* (OOBE) which is an unbiased estimate for the generalization error and can also be used for model parameter estimation. Another attempt in [Lin and Jeon, 2002] points out the relations between random forests and approximated, adaptively weighted k-nearest neighbors.

Appendix A

Publications

Contents

In this chapter all publications achieved during this PhD course are listed according to their topic in an inverse chronological order.

A.1 Object Detection

- Peter Kontschieder, Samuel Rota Bulò, Antonio Criminisi, Pushmeet Kohli, Marcello Pelillo and Horst Bischof. *Context-Sensitive Decision Forests for Object Detection.* In Proc. Neural Information Processing Systems (NIPS), December 2012.

- Peter Kontschieder, Samuel Rota Bulò, Michael Donoser, Marcello Pelillo and Horst Bischof. *Evolutionary Hough Games for Coherent Object Detection.* Journal for Computer Vision and Image Understanding (CVIU), November 2012.

- Peter Kontschieder, Hayko Riemenschneider, Michael Donoser, and Horst Bischof. *Discriminative Learning of Contour Fragments for Object Detection.* In Proc. British Machine Vision Conf. (BMVC), September 2011.

- Peter Kontschieder, Michael Donoser, Horst Bischof, Johannes Kritzinger, and Wolfgang Bauer. *Detecting Paper Fibre Cross Sections in Microtomy Images.* In Proc. Intern. Conf. on Pattern Recognition (ICPR), August 2010.

A.2 Semantic Image Labelling

- Peter Kontschieder, Pushmeet Kohli, Jamie Shotton, and Antonio Criminisi. *GeoFF: Geodesic Forests for Learning Coupled Predictors.* In Proc. Intern. Conf. on Computer Vision and Pattern Recognition (CVPR), June 2013.

- Samuel Rota Bulò, Peter Kontschieder, Marcello Pelillo, and Horst Bischof. *Structured local predictors for image labelling.* In Proc. Intern. Conf. on Computer Vision and Pattern Recognition (CVPR), June 2012.

- Peter Kontschieder, Samuel Rota Bulò, Horst Bischof, and Marcello Pelillo. *Structured class-labels in random forests for semantic image labelling.* In Proc. Intern. Conf. on Computer Vision (ICCV), November 2011.

- Peter Kontschieder, Samuel Rota Bulò, Michael Donoser, Marcello Pelillo, and Horst Bischof. *Semantic image labelling as a label puzzle game.* In Proc. British Machine Vision Conf. (BMVC), September 2011.

A.3 Object Retrieval/Clustering

- Peter Kontschieder, Michael Donoser, and Horst Bischof. *Beyond Pairwise Shape Similarity Analysis*, In Proc. Asian Conf. on Computer Vision (ACCV), September 2009.

- Peter Kontschieder, Michael Donoser, and Horst Bischof. *Improving Affinity Matrices by Modified Mutual KNN-Graphs.* 33rd Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM), May 2009.

A.4 Tracking by Detection / Augmented Reality

- Michael Donoser, Peter Kontschieder, and Horst Bischof. *Robust Planar Target Tracking and Pose Estimation from a Single Concavity.* In Proc. Intern. Symposium on Mixed and Augmented Reality (ISMAR), October 2011.

- Peter Kontschieder, Michael Donoser, and Horst Bischof. *MSER Templates for 3D Pose Tracking.* 34th Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM), May 2010.

Bibliography

[Ima, 2012] (2012). `http://www.image-net.org/challenges/LSVRC/2012/`.

[LFW, 2013] (2013). `http://vis-www.cs.umass.edu/lfw/`.

[VOC, 2013] (2013). `http://www.pascal-network.org/challenges/VOC/`.

[Abend et al., 1965] Abend, K., Harley, T. J., and Kanal, L. N. (1965). Classification of binary random patterns. *IEEE Transactions on Information Theory*, 11(4):538–544.

[Agarwal et al., 2004] Agarwal, S., Awan, A., and Roth., D. (2004). Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26:1475–1490.

[Albarelli et al., 2009] Albarelli, A., Torsello, A., Rota Bulò, S., and Pelillo, M. (2009). Matching as a non-cooperative game. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Amit and Geman, 1997] Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588.

[Andriluka et al., 2008] Andriluka, M., Roth, S., and Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Avidan, 2006] Avidan, S. (2006). Spatialboost: Adding spatial reasoning to adaboost. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Bai et al., 2009] Bai, X., Li, Q., Latecki, L. J., Liu, W., and Tu, Z. (2009). Shape band: A deformable object detection approach. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

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

[Bar, 2004] Bar, M. (2004). Visual objects in context. *Nat. Rev. Neurosci.*, 5:617–629.

[Barinova et al., 2010] Barinova, O., Lempitsky, V., and Kohli, P. (2010). On detection of multiple object instances using hough transforms. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24:509–522.

[Biederman, 1972] Biederman, I. (1972). Perceiving real-world scenes. *Science*.

[Biederman and Ju, 1988] Biederman, I. and Ju, G. (1988). Surface vs. edge-based determinants of visual recognition. *Cognitive Psychology*, 20:38–64.

[Blake et al., 2011] Blake, A., Kohli, P., and Rother, C. (2011). *Markov Random Fields for Vision and Image Processing*. The MIT Press.

[Bosch et al., 2007] Bosch, A., Zisserman, A., and Muñoz, X. (2007). Image classification using random forests and ferns. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Boykov and Jolly, 2001] Boykov, Y. and Jolly, M. P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Boykov et al., 2001] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

[Breiman, 1996a] Breiman, L. (1996a). Bagging predictors. In *Machine Learning (ML)*.

[Breiman, 1996b] Breiman, L. (1996b). Out-of-bag estimation. Technical report, University of Berkeley, California.

[Breiman, 1999] Breiman, L. (1999). Random forests. Technical report, UC Berkeley.

[Breiman, 2001] Breiman, L. (2001). Random forests. In *Machine Learning (ML)*.

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.

[Brostow et al., 2008] Brostow, G. J., Shotton, J., Fauqueur, J., and Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Carbonetto et al., 2004] Carbonetto, P., de Freitas, N., and Barnard, K. (2004). A statistical model for general contextual object recognition. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Caruana et al., 2008] Caruana, R., Karampatziakis, N., and Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In *Proc. Intern. Conf. on Machine Learning (ICML)*.

[Chen et al., 2008] Chen, L., Feris, R., and Turk, M. (2008). Efficient partial shape matching using smith-waterman algorithm. In *Proc. of NORDIA workshop at CVPR*.

[Chow, 1962] Chow, C. K. (1962). A recognition method using neighbor dependence. *IRE Trans. on Electronic Computer*, 11:683–690.

[Cohen, 1986] Cohen, S. (1986). Knowledge and context. *The Journal of Philosophy*, 83(10):574–583.

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

[Cover and Thomas, 2006] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory, Second Edition*. John Wiley & Sons.

[Criminisi et al., 2008] Criminisi, A., Sharp, T., and Blake, A. (2008). GeoS: Geodesic image segmentation. In *Proc. European Conf. on Computer Vision (ECCV)*. Springer.

[Criminisi et al., 2011] Criminisi, A., Sharp, T., Rother, C., and Perez, P. (2011). Geodesic image and video editing. *Proc. Intern. Conf. and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

[Criminisi and Shotton, 2013] Criminisi, A. and Shotton, J. (2013). *Decision Forests for Computer Vision and Medical Image Analysis*. Springer.

[Criminisi et al., 2012] Criminisi, A., Shotton, J., and Konukoglu, E. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. In *Foundations and Trends in Computer Graphics and Vision*, volume 7, pages 81–227.

[Criminisi et al., 2010] Criminisi, A., Shotton, J., Robertson, D., and Konukoglu, E. (2010). Regression forests for efficient anatomy detection and localization in CT studies. In *MICCAI Workshop on Medical Computer Vision: Recognition Techniques and Applications in Medical Imaging*.

[Dollár et al., 2010] Dollár, P., Welinder, P., and Perona, P. (2010). Cascaded pose regression. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Donoser et al., 2009] Donoser, M., Riemenschneider, H., and Bischof, H. (2009). Efficient partial shape matching of outer contours. In *Proc. Asian Conf. on Computer Vision (ACCV)*.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *Intern. Journal of Comput Vision (IJCV).*

[Fanelli et al., 2011a] Fanelli, G., Gall, J., and Gool, L. V. (2011a). Real time head pose estimation with random regression forests. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR).*

[Fanelli et al., 2011b] Fanelli, G., Weise, T., Gall, J., and Gool, L. V. (2011b). Real time head pose estimation from consumer depth cameras. In *Proc. DAGM Symposium (DAGM).*

[Felzenszwalb et al., 2010] Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*

[Fergus et al., 2003] Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 264–271.

[Ferrari et al., 2008] Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*

[Ferrari et al., 2007] Ferrari, V., Jurie, F., and Schmid, C. (2007). Accurate object detections with deformable shape models learnt from images. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR).*

[Ferrari et al., 2009] Ferrari, V., Jurie, F., and Schmid, C. (2009). From images to shape models for object detection. In *Intern. Journal of Comput Vision (IJCV).*

[Ferrari et al., 2006] Ferrari, V., Tuytelaars, T., and Gool, L. V. (2006). Object detection by contour segment networks. In *Proc. European Conf. on Computer Vision (ECCV)*, volume 3.

[Fischler and Elschlager, 1973] Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Trans. on Computers*, 100(22):67–92.

[Freund, 1995] Freund, Y. (1995). Boosting a weak learning algorithm by majority. In *Information and Computation*, pages 256–285.

[Friedman, 1977] Friedman, J. H. (1977). A recursive partitioning decision rule for nonparametric classification. In *IEEE Transactions on Computers*, pages 404–408.

[Fröhlich et al., 2012] Fröhlich, B., Rodner, E., and Denzler, J. (2012). As time goes by - anytime semantic segmentation with iterative context forests. In *Proc. DAGM Symposium (DAGM).*

[Gall and Lempitsky, 2009] Gall, J. and Lempitsky, V. (2009). Class-specific hough forests for object detection. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Gall et al., 2011] Gall, J., Yao, A., Razavi, N., Van Gool, L., and Lempitsky, V. (2011). Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

[Geman and Geman, 1984] Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 6:721–741.

[Geurts et al., 2006] Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning (ML)*.

[Girshick et al., 2011] Girshick, R., Shotton, J., Kohli, P., Criminisi, A., and Fitzgibbon, A. (2011). Efficient regression of general-activity human poses from depth images. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Glocker et al., 2012] Glocker, B., Pauly, O., Konukoglu, E., and Criminisi, A. (2012). Joint classification-regression forests for spatially structured multi-object segmentation. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Godec et al., 2010] Godec, M., Leistner, C., Saffari, A., and Bischof, H. (2010). On-line random naive bayes for tracking. In *Proc. Intern. Conf. on Pattern Recognition (ICPR)*.

[Gonfaus et al., 2010] Gonfaus, J. M., Boix, X., van de Weijer, J., Bagdanov, A. D., Serrat, J., and Gonzalez, J. (2010). Harmony potentials for joint classification and segmentation. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Grauman and Darrell, 2005] Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Gulshan et al., 2010] Gulshan, V., Rother, C., Criminisi, A., Blake, A., and Zisserman, A. (2010). Geodesic star convexity for interactive image segmentation. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Hanson and Riseman, 1978] Hanson, A. and Riseman, E. (1978). Visions: a computer vision system for interpreting scenes. *Computer Vision Systems*, pages 303–334.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning*. Springer.

[He et al., 2004a] He, X., Zemel, R. S., and Carreira-Perpiñán, M. A. (2004a). Multiscale conditional random fields for image labeling. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2.

[He et al., 2004b] He, X., Zemel, R. S., and Carreira-Perpinan, M. A. (2004b). Multiscale conditional random fields for image labeling. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

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

[Hummel and Zucker, 1983] Hummel, R. A. and Zucker, S. W. (1983). On the foundations of relaxation labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 5(3):267–287.

[Hunt et al., 1966] Hunt, E. B., Marin, J., and Stone, P. J. (1966). *Experiments in Induction*. Academic Press.

[Hyafil and Rivest, 1976] Hyafil, L. and Rivest, R. L. (1976). Constructing optimal binary decision trees is NP-complete. In *Information Processing Letters*, 5, pages 15–17.

[Jancsary et al., 2012] Jancsary, J., Nowozin, S., Sharp, T., and Rother, C. (2012). Regression tree fields - an efficient, non-parametric approach to image labeling problems. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Jurie and Schmid, 2004] Jurie, F. and Schmid, C. (2004). Scale-invariant shape features for recognition of object categories. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Kluckner et al., 2009] Kluckner, S., Mauthner, T., Roth, P. M., and Bischof, H. (2009). Semantic image classification using consistent regions and individual context. In *Proc. British Machine Vision Conf. (BMVC)*.

[Kolmogorov, 2006] Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(10).

[Kononenko et al., 1984] Kononenko, I., Bratko, I., and Roskar, E. (1984). Experiments in automatic learning of medical diagnostic rules. Technical report, Jozef Stefan Institute, Ljubljana.

[Kontschieder et al., 2011] Kontschieder, P., Rota Bulò, S., Bischof, H., and Pelillo, M. (2011). Structured class-labels in random forests for semantic image labelling. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Kontschieder et al., 2012] Kontschieder, P., Rota Bulò, S., Criminisi, A., Kohli, P., Pelillo, M., and Bischof, H. (2012). Context-sensitive decision forests for object detection. In *Proc. Neural Information Processing Systems (NIPS)*.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proc. Neural Information Processing Systems (NIPS)*.

[Kumar and Hebert, 2005] Kumar, S. and Hebert, M. (2005). A hierarchical field framework for unified context-based classification. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Ladicky et al., 2010a] Ladicky, L., Russell, C., Kohli, P., and Torr, P. (2010a). Graph cut based inference with co-occurrence statistics. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Ladicky et al., 2010b] Ladicky, L., Sturgess, P., Alahari, K., Russell, C., and Torr, P. (2010b). What, where & how many? Combining object detectors and CRFs. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. Intern. Conf. on Machine Learning (ICML)*.

[Lampert et al., 2008] Lampert, C., Blaschko, M., and Hofmann, T. (2008). Beyond sliding windows: Object localization by efficient subwindow search. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Lazebnik et al., 2006] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178.

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

[Leibe et al., 2008] Leibe, B., Leonardis, A., and Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *Intern. Journal of Comput Vision (IJCV)*.

[Leistner et al., 2009] Leistner, C., Saffari, A., Santner, J., and Bischof, H. (2009). Semi-supervised random forests. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Leordeanu et al., 2007] Leordeanu, M., Hebert, M., and Sukthankar, R. (2007). Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Lepetit and Fua, 2006] Lepetit, V. and Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28.

[Li et al., 2010] Li, F., Carreira, J., and Sminchisescu, C. (2010). Object recognition as ranking holistic figure-ground hypotheses. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Li, 1995] Li, S. Z. (1995). *Markov random field modeling in computer vision*. Springer-Verlag, London, UK, UK.

[Lim et al., 2013] Lim, J. J., Zitnick, C. L., and Dollar, P. (2013). Sketch tokens: A learned mid-level representation for contour and object detection. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Lin and Jeon, 2002] Lin, Y. and Jeon, Y. (2002). Random forests and adaptive nearest neighbors. Technical report, University of Wisconsin.

[Liu and Yan, 2010] Liu, H. and Yan, S. (2010). Common visual pattern discovery via spatially coherent correspondences. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Lu et al., 2009] Lu, C., Latecki, L. J., Adluru, N., Yang, X., and Ling, H. (2009). Shape guided contour grouping with particle filters. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Maji and Malik, 2009] Maji, S. and Malik, J. (2009). Object detection using a max-margin hough transform. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Marée et al., 2005] Marée, R., Geurts, P., Piater, J., and Wehenkel, L. (2005). Random subwindows for robust image classification. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Martin et al., 2004] Martin, D. R., Fowlkes, C. C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(10):1615–1630.

[Montillo et al., 2011] Montillo, A., Shotton, J., Winn, J., Iglesias, J. E., Metaxas, D., and Criminisi, A. (2011). Entangled decision forests and their application for semantic segmentation of CT images. In *Proc. Intern. Conf. on Information Processing in Medical Imaging*.

[Montillo et al., 2013] Montillo, A., Tu, J., Shotton, J., Winn, J., Iglesias, J. E., Metaxas, D. N., and Criminisi, A. (2013). Entanglement and differentiable information gain maximization. In *Decision Forests for Computer Vision and Medical Image Analysis*. Springer.

[Munoz et al., 2010] Munoz, D., Bagnell, J. A., and Hebert, M. (2010). Stacked hierarchical labeling. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Murthy and Salzberg, 1995] Murthy, S. K. and Salzberg, S. (1995). Decision tree induction: How effective is the greedy heuristic? In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 222–227.

[Nowozin, 2012] Nowozin, S. (2012). Improved information gain estimates for decision tree induction. In *Proc. Intern. Conf. on Machine Learning (ICML)*.

[Nowozin and Lampert, 2011] Nowozin, S. and Lampert, C. H. (2011). Structured learning and prediction in computer vision. In *Foundations and Trends in Computer Graphics and Vision*.

[Nowozin et al., 2011] Nowozin, S., Rother, C., Bagon, S., Sharp, T., Yao, B., and Kohli, P. (2011). Decision tree fields. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Okada, 2009] Okada, R. (2009). Discriminative generalized hough transform for object detection. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Ommer and Malik, 2009] Ommer, B. and Malik, J. (2009). Multi-scale object detection by clustering lines. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Opelt et al., 2006] Opelt, A., Pinz, A., and Zisserman, A. (2006). A boundary-fragment-model for object detection. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Özuysal et al., 2007] Özuysal, M., Fua, P., and Lepetit, V. (2007). Fast keypoint recognition in ten lines of code. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Patterson and Niblett, 1983] Patterson, A. and Niblett, T. (1983). ACLS user manual. *MIRU*.

[Pavan and Pelillo, 2007] Pavan, M. and Pelillo, M. (2007). Dominant sets and pairwise clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(1):167–172.

[Payet and Todorovic, 2010] Payet, N. and Todorovic, S. (2010). From a set of shapes to object discovery. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Payet and Todorovic, 2012] Payet, N. and Todorovic, S. (2012). Hough forest random field for object recognition and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

[Pelillo, 1997] Pelillo, M. (1997). The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision (JMIV)*, pages 309–323.

[Pelillo and Refice, 1994] Pelillo, M. and Refice, M. (1994). Learning compatibility coefficients for relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 16(9):933–945.

[Porikli et al., 2006] Porikli, F., Tuzel, O., and Meer, P. (2006). Covariance tracking using model update based on lie algebra. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Price, 2008] Price, A. W. (2008). *Contextuality in Practical Reason*. Oxford, UK.

[Quinlan, 1979] Quinlan, J. (1979). Discovering rules by induction from large collections of examples. In *Expert systems in the microelectronic age*.

[Quinlan, 1983] Quinlan, J. (1983). Learning efficient classification procedures and their application to chess endgames. In *Machine learning: An artificial intelligence approach*.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning (ML)*.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5. Programs for Machine Learning*. Morgan Kaufmann Publishers.

[Rabinovich et al., 2007] Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. (2007). Objects in context. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Ravishankar et al., 2008] Ravishankar, S., Jain, A., and Mittal, A. (2008). Multi-stage contour based detection of deformable objects. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Razavi et al., 2011] Razavi, N., Gall, J., and Van Gool, L. (2011). Scalable multi-class object detection. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Ren et al., 2005] Ren, X., Berg, A. C., and Malik, J. (2005). Recovering human body configurations using pairwise constraints between parts. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Riemenschneider et al., 2010] Riemenschneider, H., Donoser, M., and Bischof, H. (2010). Using partial edge contour matches for efficient object category localization. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Riemenschneider et al., 2012] Riemenschneider, H., Sternig, S., Donoser, M., Roth, P. M., and Bischof, H. (2012). Hough regions for joining instance localization and segmentation. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Rosenfeld et al., 1976] Rosenfeld, A., Hummel, R., and Zucker, S. W. (1976). Scene labeling by relaxation operations. *IEEE Trans. Syst. Man & Cybern.*, 6:420–433.

[Ross et al., 2011] Ross, S., Munoz, D., Hebert, M., and Bagnell, J. A. (2011). Learning message-passing inference machines for structured prediction. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Rota Bulò and Bomze, 2011] Rota Bulò, S. and Bomze, I. M. (2011). Infection and immunization: a new class of evolutionary game dynamics. *Games and Economic Behaviour*, 71:193–211.

[Rota Bulò et al., 2012] Rota Bulò, S., Kontschieder, P., Pelillo, M., and Bischof, H. (2012). Structured local predictors for image labelling. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Rota Bulò et al., 2011] Rota Bulò, S., Pelillo, M., and Bomze, I. M. (2011). Graph-based quadratic optimization: A fast evolutionary approach. *Computer Vision and Image Understanding (CVIU)*, pages 984–995.

[Russell et al., 2007] Russell, B. C., Torralba, A., Liu, C., Fergus, R., and Freeman, W. T. (2007). Object recognition by scene alignment. In *Proc. Neural Information Processing Systems (NIPS)*.

[Saffari et al., 2009] Saffari, A., Leistner, C., Santner, J., Godec, M., and Bischof, H. (2009). On-line random forests. In *ICCV Workshop on on-line learning for computer vision*.

[Schapire, 1990] Schapire, R. E. (1990). The strength of weak learnability. In *Machine Learning (ML)*.

[Schulter et al., 2013] Schulter, S., Wohlhart, P., Leistner, C., Saffari, A., Roth, P. M., and Bischof, H. (2013). Alternating decision forests. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Sharp, 2008] Sharp, T. (2008). Implementing decision trees and forests on a GPU. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Shepherd, 1983] Shepherd, B. A. (1983). An appraisal of a decision tree approach to image classification. In *International Joint Conference on Artificial Intelligence*, pages 473–475.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

[Shinghal et al., 2003] Shinghal, A., Luo, J., and Zhu, W. (2003). Probabilistic spatial context models for scene content understanding. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Shotton, 2007] Shotton, J. (2007). *Contour and Texture for Visual Recognition of Object Categories*. PhD thesis, University of Cambridge.

[Shotton et al., 2005] Shotton, J., Blake, A., and Cipolla, R. (2005). Contour-based learning for object detection. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Shotton et al., 2008a] Shotton, J., Blake, A., and Cipolla, R. (2008a). Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(7):1270–1281.

[Shotton et al., 2011] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from a single depth image. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Shotton et al., 2012] Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., and Blake, A. (2012). Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

[Shotton et al., 2008b] Shotton, J., Johnson, M., and Cipolla, R. (2008b). Semantic texton forests for image categorization and segmentation. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Shotton et al., 2006] Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Shotton et al., 2007] Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2007). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Intern. Journal of Comput Vision (IJCV)*, 81.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Proc. Intern. Conf. on Computer Vision (ICCV)*.

[Smith et al., 2009] Smith, K., Carleton, A., and Lepetit, V. (2009). Fast ray features for learning irregular shapes. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR).*

[Srinivasan et al., 2010] Srinivasan, P., Zhu, Q., and Shi, J. (2010). Many-to-one contour matching for describing and discriminating object shape. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR).*

[Sturgess et al., 2009] Sturgess, P., Alahari, K., Ladicky, L., and Torr, P. (2009). Combining appearance and structure from motion features for road scene understanding. In *Proc. British Machine Vision Conf. (BMVC).*

[Tarlow and Zemel, 2012] Tarlow, D. and Zemel, R. S. (2012). Structured output learning with high order loss functions. In *Intern. Conf. on Artificial Intelligence and Statistics.*

[Torralba, 2003] Torralba, A. (2003). Contextual priming for object detection. *Intern. Journal of Comput Vision (IJCV)*, 53(2):153–167.

[Torralba et al., 2004] Torralba, A., Murphy, K., and Freeman, W. (2004). Contextual models for object detection using boosted random fields. In *Proc. Neural Information Processing Systems (NIPS).*

[Torralba et al., 2005] Torralba, A., Murphy, K. P., and Freeman, W. T. (2005). Contextual models for object detection using boosted random fields. In *Proc. Neural Information Processing Systems (NIPS).*

[Torsello et al., 2006] Torsello, A., Rota Bulò, S., and Pelillo, M. (2006). Grouping with asymmetric affinities: a game-theoretic perspective. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 292–299.

[Toshev et al., 2010] Toshev, A., Taskar, B., and Daniilidis, K. (2010). Object detection via boundary structure segmentation. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR).*

[Toussaint, 1978] Toussaint, G. T. (1978). The use of context in pattern recognition. *Pattern Recognition (PR)*, 10:189–204.

[Tsochantaridis et al., 2004] Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector learning for interdependent and structured output spaces. In *Proc. Intern. Conf. on Machine Learning (ICML).*

[Tu, 2008] Tu, Z. (2008). Auto-context and its application to high-level vision tasks. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR).*

[Tu and Bai, 2010] Tu, Z. and Bai, X. (2010). Auto-context and its application to high-level vision tasks and 3D brain image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(10).

[Verbeek and Triggs, 2008] Verbeek, J. and Triggs, B. (2008). Scene segmentation with crfs learned from partially labeled images. In *Proc. Neural Information Processing Systems (NIPS)*.

[Villamizar et al., 2012] Villamizar, M., Garrell, A., Sanfeliu, A., and Moreno-Noguer, F. (2012). Online human-assisted learning using random ferns. In *Proc. Intern. Conf. on Pattern Recognition (ICPR)*.

[Viola and Jones, 2002] Viola, P. and Jones, M. (2002). Robust real-time object detection. *Intern. Journal of Comput Vision (IJCV)*.

[Weibull, 1995] Weibull, J. W. (1995). *Evolutionary Game Theory*. MIT Press.

[Wiltsche et al., 2005] Wiltsche, M., Donoser, M., Bauer, W., and Bischof, H. (2005). A new slice-based concept for 3d paper structure analysis applied to spatial coating layer formation. In *In Proc. of the 13th Fundamental Paper Research Symposium*.

[Winn and Shotton, 2006] Winn, J. M. and Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Wolpert, 1992] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5.

[Wolpert, 1996] Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. In *Neural Computation (NC)*, pages 1341–1390.

[Xu et al., 2009] Xu, C., Liu, J., and Tang, X. (2009). 2D shape matching by contour flexibility. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, volume 31.

[Yakimovsky and Feldman, 1973] Yakimovsky, Y. and Feldman, J. (1973). A semantics-based decision theory region analyzer. In *3rd international joint conference on artificial intelligence*, pages 580–588.

[Yang et al., 2011] Yang, X., Adluru, N., and Latecki, L. J. (2011). Particle filter with state permutations for solving image jigsaw puzzles. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Yang et al., 2012] Yang, Y., Li, Z., Zhang, L., Murphy, C., Hoeve, J., and Jiang, H. (2012). Local label descriptor for example based semantic image labeling. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Yao et al., 2010] Yao, A., Gall, J., and van Gool, L. (2010). A hough transform-based voting framework for action recognition. In *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[Yarlagadda et al., 2010] Yarlagadda, P., Monroy, A., and Ommer, B. (2010). Voting by grouping dependent parts. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Zhou, 2012] Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC.

[Zhu et al., 2008a] Zhu, L., Chen, Y., Lin, Y., Lin, C., and Yuille, A. (2008a). Recursive segmentation and recognition templates for 2d parsing. In *Proc. Neural Information Processing Systems (NIPS)*.

[Zhu et al., 2008b] Zhu, Q., Wang, L., Wu, Y., and Shi, J. (2008b). Contour context selection for object detection: A set-to-set contour matching approach. In *Proc. European Conf. on Computer Vision (ECCV)*.

[Zusne, 1970] Zusne, L. (1970). *Visual Perception of Form*. Academic Press, New York.