Graz University of Technology

Institute for Computer Graphics and Vision

Dissertation

Object Detection from Aerial Image

NGUYEN Thi Thuy

Graz, Austria, September 2009

1. Supervisor and Examiner

Prof. Dr. Horst Bischof

2. Examiner

Prof. Dr. Vaclav Hlavac

To my family.

Everything should be made as simple as possible, but not simpler.

A. Einstein

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)

Acknowledgments

Firstly, I would like to express my best gratitude to my supervisor, Professor Horst Bischof, for giving me an opportunity of doing PhD. at the Institute for Computer Graphics and Vision at the Graz University of Technology, Austria. His useful advices, suggestions and support guided me through the years of research.

Both the Institute for Computer Graphics and the Institute for Theoretical Computer Science, at which I was working, are filled with stimulating, helpful, and affable people. It would take this thesis to thank individually everyone that has helped me through the course. I do wish to extend my thanks to a few individuals: The work in Chapter 2 has been created with fruitful collaboration with Helmut Grabner. My work can not be finished in a practical and meaningful way without valued data of aerial images from Microsoft Vexcel where Babara Gruber and Stefan Kluckner are directly involved. Thanks to Amir Saffari for his side-yet-useful discussion and being here and there through some of my up and down time here in Graz. My thanks are also given to members of the Institute for Computer Graphics and Vision for their support and enthusiasm with pleasure working environment.

I would like to extend my thanks to Professor Wolfgang Mass and a group of colleagues at the Institute for Theoretical Computer Science for their help at my very beginning time in Graz. My knowledge of Machine learning was initiated there. My special thanks to Professor Vaclav Hlavac for providing me pleasant time visiting the Center for Machine Perception at Czech Technical University in Praha, and for his constructive and useful advices at the final step of writing this thesis.

I am grateful to OeAD for their financial support and useful help for almost my studying time here in Austria. Without which my study would not have been possible.

Last but not least, my heartfelt thanks to my family and close friends for their being here and there with encouragement, support, and love all the time.

Abstract

This thesis aims at developing learning systems for automated detection of objects from aerial images. The problem is challenging not only because of the intricacy of complex objects in natural scene, but also the computational demand to process huge image data. The interested objects can be roughly divided into two main categories: Objects with regular shape, well-located, such as cars, trucks; and, objects with complex shape, spread over the scene, such as buildings, road net. To this end, the thesis concerns two approaches for the detection of objects of the two categories.

(i) We develop a learning framework based on the online boosting method for detecting of regular objects, e.g. cars. The main contribution is an efficient framework for automatic detection of cars from large scale aerial images. The efficient feature representation with an interactive on-line boosting learning allows the detector to be trained and improved effectively. Further improvements can be obtained by a new active learning procedure.

(ii) We propose a novel probabilistic model, named HpCRF: hierarchical pseudo-conditional random field, and a learning algorithm to train the model for the detection and the segmentation of complex objects, i.e. buildings. The idea is to treat different feature types in separated processes, exploit their discriminative potentials and interactions, and then integrate them into an unified hierarchical probabilistic model. This aims at to fully exploit the power of each individual feature type and to leverage the performance by using context at higher levels. Our approach works at the pixel level to obtain a pixel-wise accuracy of the object class. Learning and inference are effective, general and straightforward. The approach can be applied to a number of learning tasks.

An extensive set of experiments on real world data sets of large aerial images have been conducted for both approaches. The experiments show the applicability and the benefit of our methods over traditional state-of-the-art approaches.

Keywords: Computer vision, Machine learning, Object detection/recognition, Car detection, Building detection, Aerial photo interpretation, Semantic segmentation, Supervised learning, Online boosting, Active learning, Conditional random field, Hierarchical probabilistic model.

Kurzfassung

Diese Arbeit beschäftigt sich mit der Entwicklung von Lernsystemen für eine automatische Detektion von Objekten in Luftbildern. Die Vielfalt an komplexen Objekten in natürlichen Bildszenen und die enormen Datenmengen stellen sehr hohe Ansprüche an die Problemlösungen. Objekte von Interesse können grob in zwei Kategorien eingeteilt werden. Objekte mit einer kompakten Form wie zum Beispiel Autos und LKWs, und komplexe, form-variable Objekte wie Gebäude und Straßenzüge. In dieser Arbeit werden daher zwei Methoden vorgestellt: (i) Die erste Methode befasst sich mit der Entwicklung einer automatischen Lernumgebung, basierend auf Online-Boosting, für eine effiziente Detektion von Objekten wie Autos in großflächigen Luftbildern. Eine kombinierte Merkmalsrepräsentation und die interaktive Lernmethode erlaubt eine effektive Verbesserung des Klassifikators durch eine aktive Trainingsprozedur. (ii) Weiters wird ein hierarchisch-probabilistisches Model, HpCRF genannt, vorgestellt. Es wird ein neuer Lernalgorithmus erläutert, der Detektion und Segmentierung von komplexen Objekten wie Gebäude erlaubt. Die Idee basiert auf einer vorerst getrennten Behandlung bzw. Klassifikation von diverser Merkmalstypen um diskriminative Eigenschaften besser nützen zu können. Eine Kombination erfolgt dann im probabilistischen Model. Dadurch werden die individuellen Merkmale effektiver genützt und das Ergebnis durch Verwendung von Kontext auf einem höheren Level verbessert. Die vorgestellte Methode wird auf der Pixel- Ebene berechnet und evaluiert, zudem zeichnet sie sich durch Effektivität und Einfachheit aus. Für die Experimente werden reale, großflächige Luftbilder verwendet um die vorgestellten Methoden ausführlich zu evaluieren. Die Versuche zeigen die Anwendbarkeit und die Vorteile gegenüber herkömmlichen State-of-the-Art Methoden.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Aerial Image Interpretation

Aerial Photograph

Aerial photography has come and made its contribution to our society since over 100 years with many discoveries and inventions*. It is considered as a combination of art, science and technology, which strives to derive locations, shapes and other information of objects from images with the best worthwhile values.

For a long stage, traditional film cameras have been used for photography. In the last decade, digital aerial cameras are largely replacing analogue cameras, for which the transition from film to digital images has been made. This makes dramatic changes in the field of image acquisition as well as the photogrammetric work-flow for aerial data analysis and applications.

*http://www.papainternational.org/history.html, Aug. 2009.

The main advanced properties of digital cameras can be summarized as: the ability to produce a higher overlap, up to 90% along track without additional expenses for film; the multispectral sensing capability; the unchanged digital workflow with its inherent properties; the producing of much higher level of automation in photogrametric data analysis. The advantages of highly image redundancy and thus robust of analyzing and maximizing the automatic work flow have made the "paradign shift" in aerial photogrametry [85, 86]. As a general consequence, "the quality of the image products have greatly increased and the full automation of mapping processes could be possible in the near future" [135].

Recently, a variety of aerial digital imaging systems are in operation and the camera system is still evolving*. The camera operation provides large scale digital aerial images in multiple channels at an efficient cost. One of such products is the *UltracamD* from Microsoft Vexcel† (former Vexcel Imaging).

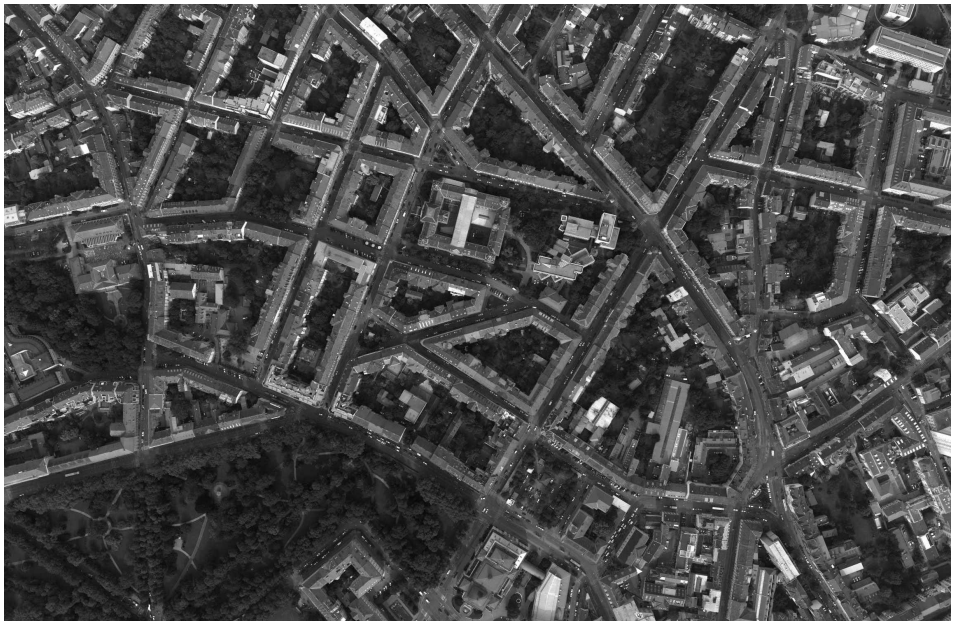
The camera UltracamD is designed with its multi-spectral capability and multi-head, i.e. nine CCD sensors, which offers simultaneously high resolution of panchromatic channel as well as lower resolution RGB and NIR channels. Therefore, the camera can deliver large scale image format of 11500 pixels across-track and 7500 pixels along-track for panchromatic images, and at a size of 3680 by 2400 pixels for four other channels (red, green, blue and NIR) [84]. Recently, UltraCamX has been introduced with more advanced technology and capability of producing 14430 across track and 9420 along track pixels†.

The contribution in this thesis has been made mainly based on the image data provided by the UltraCamD camera from Microsoft Photogrammetry. The used image data comprise the panchromatic high resolution images as well as the multispectral images. Two typical images of the data set of Graz city are shown in Figure 1.1.

Aerial Photograph Interpretation (API)

Aerial photograph interpretation (API, also termed "Aerial image interpretation") can be understood in several slightly different ways, such as:
(1)- The act of examining aerial photographs for the purpose of identifying objects and judging their significance‡.
(2)- The process of location, recognition, identification, and description of objects, activities, and terrain represented on imagery§.

*http://www.aerial-survey-base.com/cameras.html, Jul. 2009.
†http://www.microsoft.com/ultracam/default.mspx, Jul. 2009.
‡http://www.r-s-c-c.org/rscc/v1m2.html, Jul. 2009.
§http://www.thefreedictionary.com/imagery+interpretation, Jul. 2009.



(a)

(b)

Figure 1.1: An example of aerial image captured by *UltraCamD*: a Panchromatic image (a) and an RGB image (b).

(3)- The process of studying and gathering the information required to identify the various cultural and natural features is called photo interpretation[*].

In general, API can be considered as the process of analyzing and understanding meaningful objects containing in the images, which is one of the most important problem in computer vision and in photogrammetry.

As an object recognition task, the API process has to take into account (principle) visual characteristic of objects, which allow us to differentiate them. These characteristics include fundamental recognition elements such as color, texture, size, shape, context, etc. On one level, such interpretation consists of recognizing objects of certain categories (e.g. houses, cars, people) and their location within the scene. At higher level, the goal is to utilize the recognized objects for various practical applications, such as cartography, land cover monitoring, or 3D city modeling.

Since the first aerial photograph has been taken over one hundred years ago, technology has changed and allowed for clearer and more detailed photographs to be taken and interpreted. As an evidence, the *web* link of Aero-Data Corp. (http://www.aero-data.com/photointerpretation.html, Jul. 2009) gives an interesting example of some aerial photographs and the image interpretation from a historical study of a site. It shows how a particular area changed over time from a refinery in the 1930's, to a residential area in the 1990's.

Literature covers a broad spectrum of topics where API plays important role, from giving important information for management strategies of ecosystems [30, 82], to address social, economic and human induced environmental changes [3]. The Landsat satellite program set up by NASA[¶] since 1970s has provided images for scientific research and discovery, and gave important information to farmers, scientists, and policy makers. Many of maps have been created from the images that facilitate the research work, decision making as well as education.

Recent development in aerial imaging technology has led to not only improvement of traditional applications (e.g. mapping performance), but also opening up a number of advanced applications. Among the various growing research interests in the field, notable topics are the automatic creation of 3D models of urban spaces [205] for various applications, e.g. urban planning, Internet applications (Microsoft Virtual Earth, Google Earth).

3D city models have become one of the most important and attractive products of

[*]http://airphotos.nrcan.gc.ca, Jul. 2009.
[¶]http://landsat.gsfc.nasa.gov/about/, Jul. 2009

photogrammetry and remote sensing. For which, automated API is an essential need. Many systems have been developed and they are different in: type of input data (aerial imagery, laser scanning, or high resolution satellite images), type of image used (mono, stereo, multiple or orthoimages), object modelling and data structures [5, 110]. These advanced topics in some case only become feasible with the recent development of related fields, such as aerial imaging technology, new methods in computer vision, etc.

The scientific work of the Technical Commissions of the ISPRS (*International Society for Photogrammetry and Remote Sensing*) with many research works presented at the *ISPRS Congress 2008*[*], the organizing of workshop on *City Models, Roads and Traffic CMRT 2009*[‖], as well as various other publications [18, 66, 109, 110, 148, 200] show special interests in the field. Recent arising research activities at academic as well as at agencies/organizations show intensive demands of practical applications. Some example are: the action of the European Spatial Data Research Network (EuroSDR, *http* : *//www.eurosdr.net*) with the setting up of the COST Action TU0801[**], the development of the City Geography Markup Language CityGML[††] tools, and various commercial products of Miscrofost Vexcel (http://www.vexcel.com), Bluesky Team (www.bluesky-world.com), etc.

The problem of automatic extraction of objects from aerial and space images for Aerial Photograph Interpretation have been an active research for decades [48, 173]. In recent years, the field has been extremely advanced by the vision research community [64, 109, 125, 138]. However, the photogrammetric reconstruction of complicated objects with various appearance and shapes still faces the problems including image understanding, automatic control of levels of detail (LoD) and topology generation [49, 205]. Very recently, a comprehensive review of automated object extraction in photogrammetric computer vision by Mayer [109] has showed an arising of practical aspects for Digital Photogrammetric Workstations (DPW), especially the difficult of automated object extraction. Therefore, issues for a practical success of automated object extraction from aerial image are of important need.

In this thesis, we address the problem of automatic detection of interested objects from aerial images. The work partly facilitates the automated object extraction for the digital photogrammetric workflow. We approach the solution by developing novel object recog-

[*] http://www.isprs.org/congresses/beijing2008, Jul. 2009.
[‖] http://www.cmrt09.bv.tum.de/topics.html, Jul. 2009.
[**] http://www.semcity.eu/index.html, Jul. 2009.
[††] http://www.citygml.org/, Jul. 2009.

nition frameworks using machine learning methods. A solid and most powerful machine learning method will be discussed and developed for the detection of interested object of each category. We advocate the appearance based approach and statistical modeling methods.

1.2 Motivation

Human beings can effortlessly perform object recognition better than artificial vision systems. Human visual system is of no doubt an inspiration for building object recognition system to emulate it [23, 100, 118]. However, there exist a number of solutions for industrial applications, in which the systems can recognize and localize objects much faster and reliable than human can do [45], for instance some Cognex products*.

This is especially true in our case, where the human effort is less effect, i.e. the recognition of objects, e.g. cars or buildings from aerial images. Obviously human can do the task. However, it is extremely hard to process for huge demand of image data (multiple aerial images of a city). One question could arise: how much time and labor work are needed for manually detect cars from an aerial image of a city? Moreover, detection is not just simply to **see** the objects, but it is the essential need for further processing steps in the automated photogrammetry work-flow. For example, the detection of cars for removing car object for the task of recovering road texture, see Figure 1.2. Therefore, the motivation of this work goes beyond purely theoretical research work to aim at several important real life applications.

In this thesis, we aim at developing novel computer vision systems to facilitate several tasks of API, 3D city modeling as well as other potential applications:

- **3D city modeling**

  We use the term 3D city model to refer to semantic and geometry description of object categories in urban scene. General city objects of interest include buildings, traffic network and vegetation. For a 3D modeling of a city, these objects need to be detected and extracted. The results will provide rich image description for the 3D-city models at high levels of details.

  The creation of real world models for innovative 3D applications is expensive, time consuming, even impractical if manual methods are used (e.g. Figure 1.2). This motivates our work to develop robust and efficient procedures for automatic recognition

*http://www.cognex.com/products/visiontools/patmax, Jul. 2009.

Figure 1.2: An application of the car detection: original image (top), the detected car masks (middle), and the inpainting - remove the detected cars for road texture recovery (bottom).

of objects to facilitate the automatic photogrammetry work-flow.

We claim that automatic and efficient procedures can be developed for detecting of interested objects in aerial imagery. In this work, we limit the scope to the semantic description of objects and focus on the object categorical recognition task.

- **Semantic knowledge enrichment**
  Recently, systems for automation of creating 3D city model have been developed thanks to the evolution of digital aerial photography [78, 114, 124, 199, 200, 204]. It has been argued that "the next big step forward is to replace photographic texture by an interpretation of what the texture describes, and to achieve this fully automatically" [83]. This has been also set as the main objective of the COST Action* "to semantically enrich 3D models with urban knowledge and models, so as to extend their functionality and usability in a perspective of sustainability".

  The semantic information consists of the process of assigning labels to objects in the scene and then present the objects with meaningful texture. This work is a principle task of object recognition/categorization. By doing this, we can make a further step toward the enrichment of semantic knowledge about the present of objects in aerial imagery.

- **Land use classification**
  The specific land-use classes included water, agricultural fields, man-made objects, etc. The classification of land cover information from aerial images into different land-use categories can be utilized for urban planing, agricultural and environmental applications. The *UltracamD* images with the multi-spectral and multi viewing capability to capture every visible points on the ground provide a high potential in producing the land use classification maps.

- **Change detection**
  In the last decade, 2D topographic databases have been completed in most developed countries. Most efforts in mapping issues are now dedicated to the update of such databases. The task is generally carried out manually by visual inspection of an orthophoto to detect objects to be revised, which is costly, time-consuming and tedious [171]. There is a growing need to develop tools to automate the process. The need has driven the EuroSDR‡‡ to set up a *change detection* project. Several algo-

*http://www.semcity.eu, Jul. 2009.
‡‡http://www.eurosdr.net, Jul. 2009.

rithms with state-of-the-art methods in the field have been employed. Preliminary results have been reported [21]. Despite promising results, most of the approaches shown with significant false alarm rates. Obviously, automatic solutions have to face the complexity and the diversity of building structures present in urban areas. Moreover, the used frameworks are rather complicated and less efficient.

Why the detection of cars is of interest?

For the 3D city modeling, small objects such as cars on the street are considered as noisy objects, which should be removed to recover the main characteristic texture of the road. Detection of cars is necessary for the task (see Figure 1.2).

Beside that, detection of cars can provide contextual information about the appearance of road or parking lot, which could support the process of road extraction or street layer verification. On the contrary, detection of road could give prior information about the appearance of cars.

For a traffic monitoring and management problem, cars objects are of high interest. The number of cars/vehicles, their location and density represent the essential base data for traffic flow description and modeling [180].

Why the detection of buildings is of interest?

For the reconstruction of buildings from airborne imaging (aerial image or laser scanning data), much attention was given to the reconstruction of roof forms. Many of the algorithms assume that the footprint of the building is provided, e.g. by a digital cadastral map. While this kind of data is not available in many areas, it is not up-to-date, i.e. the task of map revision requires detection of buildings (see the *Change detection* section above). The quality of building footprint extraction is still mostly unknown and the problem is not solved satisfactory [138]. A number of algorithms has been published on building detection, but neither has the meaning of "successfully detected" been investigated in detail, especially for the delineation of building boundaries at pixel level of accuracy.

Although a lots of systems have been developed, novel machine learning methods have not been fully exploited for solving the problem. There are still significant needs to develop robust and efficient systems for object recognition from aerial imagery with novel machine learning methods.

Last but not least, originally motivated by several applications of the API, we set our ambition to develop novel object recognition systems that can be applied to learning and detection other objects of the same categories in aerial images as well as in terrestrial images.

1.3 Problem Statement

The problem of object detection* concerns to determine the present of the interested object in images and their locations. It is one of the most important problems of computer vision and plays vital role in image understanding or scene interpretation. Over the years, the problem of automated objects detection from aerial image has been an active research topic.

Aerial images comprise the very complex real world scenes with objects ranging from man-made to natural ones, which make the problem of object recognition challenging. The interested objects can be mainly divided into two categories: (1) The **regular object** class includes objects with regular shape, well-located and stay in isolation (countable), e.g. cars, trucks, windows; and (2) The **complex object** class includes objects which have complex shape, spread over the scene, may stay connected, e.g. buildings, road nets. These two object categories are intrinsically different and may have different meanings to the aerial image interpretation (e.g. the detection of buildings as important object to keep for the 3D modeling process, while the detection of cars as noisy objects for the removal).

Since there is no universal detector for all kind of objects, it is necessary to develop separated systems for detecting of objects of the two categories.

In this thesis, we focus on two typical objects which represent for the two categories: cars and buildings. For the car object class, it is sufficient to detect and locate a car by determining a bounding box containing the car. Thus, a sliding window approach is suitable. We set the problem to develop an efficient framework for learning a robust classifier for car detection. For the buildings object class, since the objects appear with various size, shapes, may stay connected and spread over the scene, the sliding window approach is no longer suited. Moreover, it is of special interest to obtain accurate delineation of buildings boundaries at pixel level. Therefore, a novel probabilistic model working at pixel level for classifying the object class is needed. The key problem to address is a robust model for learning a classifier for the detection/recognition of each object class.

*In the context of this thesis, the problem of object detection is a generic object recognition problem. Thus, the terms *object detection*, *object recognition* can be used interchangeably without a confusion.

The developed models are based on state-of-the-art machine learning methods. The used features for recognizing object class are mainly basic and general visual features. Therefore, the models developed in this thesis can be extended and applied for other kind of objects of the same category either from aerial or terrestrial images.

1.4 Objectives

This thesis aims to develop learning systems for automatic detection of interested objects from aerial images. In particular, we are interested in building an efficient framework for automatic detection of cars from aerial images, and developing a novel probabilistic model for the detection and segmentation of buildings at pixel level. The systems have to meet the following issues:

- Robust to deal with vast variations, inter-class and intra-class, of object's appearances in such complex natural scene.
- General enough to be able to apply to other object classes in the same category, not only for the detection of objects in aerial images but also for terrestrial images.
- Computational efficiency to handle the huge demand of large scale aerial image data.

1.5 Approaches

To address the objectives, we use novel machine learning approaches to learn classifiers to discriminate the objects from background. The object detection problem is treated as learning problem: the system learns the object's properties from a set of training samples, then it is applied on test data to detect the object class. The learning of the object class is modeled as a binary classification problem.
In this thesis, the detection of objects of the two categories are treated as two separated problems. Therefore, two systems based on two different learning approaches will be developed.

For the detection of the regular object class, i.e. cars, a suitable way is to use a search window, sliding over the image and detect the object in a window with a classifier. To learn a car detection system, we propose a robust boosting-based system for learning an implicit model of the car object. The main goal is a high quality detection by using novel machine learning methods with an efficient training mechanism. First, we use boosting and

particularly an efficient integral image representation for fast calculation of car's features. In addition to the commonly used Haar wavelets, we employ local orientation histograms and local binary patterns as features. Second, we use a novel on-line version of Adaboost to train the detector. It performs on-line updating on the ensembles of features during the training process. By on-line training, we can update the classifier as new samples arrive, and therefore we can minimize the tedious work of hand labeling of training samples. The developed framework results in a robust and automatic car detection system from aerial images achieving high performance. The system does not require any site-model or contextual knowledge or other information influencing the appearance of cars in images.
As an extension, an efficient boosting-based active learning is proposed to combine a bootstrap procedure and a semi automatic learning process. The idea is to exploit the availability of classifier during learning to automatically label training samples and increasingly improves the classifier. This addresses the issue of further reducing labeling effort meanwhile obtaining better performance.

For the complex object class, we propose a novel probabilistic approach for learning a hierarchical random field model for the detection and segmentation of building object. The idea is to treat different object's properties in separated processes and then integrate them into a unified probabilistic model. This aims to fully exploit the power of each individual feature type and to leverage the performance by using context at higher level. First, multiple features are extracted for training different discriminative classifiers. This enables us to exploit the discriminative power of each feature type and to avoid overfitting due to correlation in the feature space. Then, a novel probabilistic model is built up based on the features and the classifier outputs. This allows to learn inter-dependencies between feature types and to integrate contextual information efficiently. At the top level, classification confidences from individual classifiers are incorporated and fused to infer the object class. The proposed system provides a simple yet efficient way to model complex object class. Learning and inference are effective, general and straight forward. It can be easily used for a number of learning tasks.

The proposed systems are then tested on huge data set of aerial images for detection of objects of the two categories. Experimental results show the applicability of our approaches to real life applications. Performance evaluations are compared with other approaches show the efficiency and the superiority of our approaches over traditional and state-of-the-art methods.

1.6 Contributions

The main contributions of the thesis can be summarized as followings.

- Develops an efficient learning framework base on online boosting method for learning and detection of regular object class, i.e. car. The system allows the use of efficient representative features, effective learning and improvement of the classifier. Robust performance is obtained with fast processing on large aerial image data. The framework does not rely on any priori knowledge of the image like a site-model or contextual information (road net or street layer), but if necessary this information can be incorporated. The framework is focused on the detection of cars, but for other objects in the category the approach is applicable.

- Extends the online boosting framework with an active learning procedure to exploit the availability of classifier during learning. This learning strategy allows to automatically label training samples "on-the-fly" for learning, which greedy improves the classifier. This addresses the issue of reducing labeling effort meanwhile obtain better performance. It gives a demonstration that active learning based on an online boosting approach trained in this manner can achieve results comparable or even outperform a framework trained in conventional manner using more labeling effort.

- Propose a novel probabilistic model, namely HpCRF - A Hierarchical Pseudo-Conditional Random Field Model, based on the theory of conditional random field (CRF) and graphical model learning. The main advantage of the proposed model is its ability to incorporate mutual dependencies among different aspects of image data as well as their spatial dependencies. The HpCRF is constructed as layered model to capture different aspects of image data and contextual information. Each layer is composed of multiple CRFs. Each CRF is responsible for a certain image property and its context, which aims at exploiting discriminative power of each feature type. The hierarchical formulation allows to integrate different feature types, their inter-dependencies and spatial context potentials in a consistent model.

- Develops a learning algorithm for the proposed HpCRF model. Learning is performed sequentially for each layer. The learning of individual CRF is performed by a pseudo strategy. Any discriminative classifier which can give probabilistic output can be employed for learning the base classifier of each CRF. Inference is done by

a non-iterative Iterated Conditional Modes (ICM) method to get the classification potentials. Learning and inference of the HpCRF model are simple yet effective and general that can be considered as a meta learning scheme and can be easily applied for a number of learning tasks.

- Demonstrates the applicability and the benefits of the proposed models in several applications. Especially the focus is on the detection of cars and the detection and segmentation of buildings from aerial images.

1.7 Outline of the Thesis

The thesis is organized as follows: The Introduction section with a discussion about aerial image interpretation and motivation for this work has been presented in Chapter 1 (this chapter). A brief review of research on object recognition in computer vision is given in the Appendix *A*.

Chapter 2 and Chapter 3 are the two main chapters of the thesis. In which two different approaches are proposed for the detection of objects of the two categories, i.e. cars and buildings. In each of the two chapters, the sketch is as follows:

(1) Introduction, related work on the detection of the object from aerial imagery.

(2) Background - theoretical issues for each method.

(3) Our proposed method, and

(4) Experiments and evaluation for each approach.

In particular: Chapter 2 is dedicated for the car detection problem. We first give an introduction and a review of related work on the car detection from aerial images. We then review the background issues concerning Online boosting and active learning. In which, Boosting learning algorithm, Online boosting, boosting for feature selection and active learning are addressed. Next, we present our framework of Online boosting based learning for car detection from aerial images. An active learning procedure is introduced after that. Experiments, evaluation of the performance of the framework on the car detection problem are then presented. Ad Discussion about the approach is also included.

Chapter 3 is dedicated to the buildings detection problem. We first review recent work and the state-of-the-art of buildings detection and extraction from aerial imagery. We then present related issues of the random field models and the role of contextual information in object detection/recognition. For which, Markov random field model, conditional random field model, hierarchical modeling structures, parameter learning

and inference methods will be discussed. Next, we present our novel HpCRF model, its components, the learning and the inference algorithm. The experiments and performance evaluation of the new model will be presented after that. In Chapter 4, after summarizing the main contributions of the thesis, we will give a discussion and an outlook for the future works.

Chapter 2

Online Boosting Learning and the Car Detection Problem

Contents

Detection of an object aims to determine the presence of object in the scene. The problem is to answer the question: how many occurrences of the object X are in the image and where are they? Our goal has been to develop a system to learn the object model from a set of training samples. The system is then applied for detection of object class in unseen images. This chapter aims to develop an object detection system that is able to automatically detect objects of interest, based on on-line boosting algorithm. We focus on a novel and robust framework for automatic detection of cars from aerial images.

Firstly, we briefly review literature work on the car detection from aerial images and introduce the work which will be presented in this chapter. Secondly, we present several issues of related methods that we based on to build our system. These include: Boosting algorithm (Adaboost), Boosting for feature selection, On-line boosting learning and On-line boosting for feature selection algorithms. Thirdly, we present two learning frameworks for the car detection problem. The main contribution is a new on-line boosting learning framework for the efficient detection of cars from large-scale aerial images. In particular,

Boosting with interactive training allows the car detector to be trained and improved efficiently without requiring of labeled training data in priori. Online interactive learning allows training samples being diversified and adjusted during training to capture the variability of the real data. After training, detection is performed by exhaustive search. For post processing, a mean shift clustering method is employed, improving the detection rate significantly. For further improvement of performance of the system and reducing hand labeling effort, an on-line boosting based active learning procedure is investigated. Finally, experimental evaluation and discussion are presented.

2.1 Introduction to the Car Detection Problem

Building an efficient and robust framework for object detection from aerial images has drawn the attention of research community in computer vision for years, e.g. [4, 58, 147, 157, 202]. However, given a large-scale aerial image with typical car and background appearance variations, robust and efficient car detection system is still a challenging problem.

Aerial images are usually taken from the vertical direction and contain a lot of objects with a complicated background of the urban scene. Although with some constraints on the viewpoint, the appearance of the cars in the image is varying widely. Cars appear as small objects, which vary in intensity and many details are not visible. Depending on the resolution a typical car has a size between 13 and 26 pixels [202]. The appearance of cars may have parts occluded by the shadow of buildings or trees, or may be dominated by the shadow of the car. Moreover, the urban scene comprises a complicated background with a variety of objects that look like cars such as windows, parts of roof, corners of streets, etc. Figure 2.1 shows a complex urban scene in a sub patch of an aerial image.

Beside that, the *UltraCamD* camera from Microsoft Vexcel can deliver large format panchromatic images as well as multi spectral images [84]. The high resolution images have a size of 11500 pixels across-track and 7500 pixels along-track. Thus, a panchromatic image has a size of 84 MB, and a RGB or NIR (near infrared) image has a size of 252 MB. These big data sets of large images need automatic and efficient methods for processing.

All these issues make it difficult to characterize the features of a car and imposes challenges in recognition of cars from aerial images. Therefore, although a lot of efforts has been made, it is still an open problem to build an efficient and robust algorithm for automatic car detection from aerial images.

Figure 2.1: Example of a complex scene in urban area of an aerial image, where cars appear as small objects

Related work

The reader is referred to Appendix *A* for an overview of related work for object recognition in general. Here we briefly highlight some work directly related to object (car) detection from aerial images using machine learning methods.

Recently, a lot of research has been dedicated to object recognition using machine learning methods, e.g. [11, 55, 133, 165]. Related work on car detection can be roughly divided into two groups of approaches according to the type of modeling: explicit and implicit models [58].

Explicit modeling uses a generic car model [58, 59, 113, 162, 202]. A car is represented as a 2D or 3D model representing the shape of cars, e.g. by a box or wire-frame represen-

tation. Prominent geometric features of cars are used on different levels of detail. In the detection stage, image features are extracted and grouped to construct structures similar to the model. Mainly used features are rectangles of car boundaries or the front windshields. Additionally, radiometric features such as intensity of shadow or color can also be employed. Car detection is done by grouping extracted image features "bottom-up" or by matching the model "top-down" to the image. The car object is considered to be detected if there is sufficient evidence for the model in the image. This approach relies mainly on geometric features such as edges, lines and areas to construct a hierarchical structure. One to the ground resolution of aerial images, in the decimeter range, the models can not be very detailed because the features would not be detectable. On the other hand, generic and simple models have the inherent danger of fitting to too many positions in the image, therefore not being discriminative enough.

In an implicit or appearance-based approach, the car model is created by example images of cars and consists of gray value or texture features. Appearance models are generated by collecting statistics over these features. For car detection in terrestrial images, some part or component based models have been proposed [11, 15, 55, 89]. The classifier architecture can be a single classifier, a combination of classifiers or a hierarchical model for classification. Support vector machines were widely used [11, 55, 133, 147, 165]. For image regions, the detection is done by computing feature vectors and classifying them according to the model features. Although these approaches have certain advantages, there also exist drawbacks: Feature calculation and classification are computational expensive. Moreover, there is a need for a huge amount of labeled data for training the detector. The training set should provide a good coverage of the space of possible appearance variations of the data. This needs a lot of time and labor to build a representative training set.

Co-training versions of the classifier have been proposed to deal with this problem for object detection and classification [1, 63, 92, 119, 153]. The on-line strategy aims at reducing the manual labeling effort and makes possible to increase variability of the training data, while progressively improving the classifier.

Most related approaches attempt to match the model with the images to detect/recognize appearances of cars. Additionally, some methods limit the search area by taking into account site-model or contextual knowledge [113, 202]. For example, cars are only searched on known roads or parking lots. No method has yet explored the power of state-of-the-art machine learning methods, such as Adaboost, for adaptively and efficiently training a car detector for large scale aerial images.

Recent years, boosting - a machine learning method - has become popular. Referring to the overview given in [159], boosting has been used for text recognition, routing, medical diagnostic, segmentation, etc. Various boosting frameworks have been developed for solving machine learning problems [29, 40, 159, 172]. Following the remarkable success of the face detector introduced by Viola and Jones in [187], boosting techniques have been widely used for different problems in the computer vision community. The detection problem is formulated as a binary classification problem, discriminating the object from the background. The learned classifier is evaluated on the whole image. In order to speed up the exhaustive search, in the classical work of [187] integral images were employed, which allow very fast computation of simple image features for object representation. Additionally, a cascade structure makes the detector simultaneously fast and accurate. This framework allows to proceed efficiently on large image data and has been successfully applied for various object detection problems.

Most of the above work uses Adaboost for the detection of objects in terrestrial images. None of them (up to our knowledge) uses boosting methods for object (car) detection from aerial images. In this chapter, we develop a robust boosting-based learning system for car detection from aerial images. The main goal is high quality detection by using novel machine learning methods with an efficient training mechanism.

First, we use boosting and particularly an efficient integral image representation for fast calculation of cars' features. In addition to the commonly used Haar wavelets [187], we employ local orientation histograms [28] and local binary patterns [127] as features.

Second, we use a novel on-line version of Adaboost to train the detector. It performs on-line updating on the ensembles of features during the training process. By on-line training, we can update the classifier as new samples arrive, and therefore we can avoid building training set in advance and minimize the tedious work of hand labeling of training samples.

The developed framework results in a robust and automatic car detection system from aerial images achieving high performance. The system is flexible since it does not require any site-model or contextual knowledge or other information influencing the appearance of cars in images. But, if necessary this information can be incorporated.

2.2 Preliminaries of Online Boosting Learning

2.2.1 Boosting algorithm: Adaboost

Ensemble method is one of the machine learning algorithms that can improve the performance of any learning algorithm by a combination of the base learning models. Boosting is one of the most well-known ensemble learning methods with well-studied theory and strong supporting experimental results. In boosting, each of the base model is been trained on different re-weighted versions of training data.

The algorithm has been analyzed carefully and tested empirically by many researchers in the community, e.g. [160]. Various variants of Boosting have been developed, e.g. AdaBoost [40], Real-Boost [40], LP-Boost [29]. We define some terms which will be used through the chapter:

- Weak classifier: A weak classifier is a learning algorithm that needs to perform just better than random guessing, i.e. for the binary classification the error rate must be less than 50%. A hypothesis generated by a weak classifier is called a weak hypothesis and denoted as $h^{weak}(\mathbf{x})$

- Strong classifier: Given a set of N weak classifiers, a strong classifier is produced by a linear combination of the weak classifiers.

We focus on a specific boosting algorithm for classification, the discrete AdaBoost (adaptive boosting) algorithm introduced by Freund and Schapire [40]. The algorithm adaptively re-weights the training samples instead of re-sampling them. The basic algorithm works as follows:
Given a training set $\mathcal{X} = \{(\mathbf{x_1}, y_1), ..., (\mathbf{x_L}, y_L) \mid \mathbf{x_i} \in \mathbf{R}^m,\ y_i \in \{-1, +1\}\}$ with positive and negative labeled samples, and an initial uniform weight distribution $w_{0,i} = \frac{1}{L}$ over the examples. Based on $\mathcal{X}$ and $w(\mathbf{x})$, a weak classifier h^{weak} is trained by applying a learning algorithm, which can be a statistical learning or a decision stump, i.e,

$$h_n = \text{argmin}_{h_n} \sum_{i=1}^{L} w_{n,i} \cdot \begin{cases} 1 & h_n(\mathbf{x_i}) \neq y_i \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

Based on the error e_n on training data $\mathcal{X}$:

$$e_n = \sum_{i=1}^{L} w_{n,i} h_n(\mathbf{x_i}) \mathbf{y_i}, \tag{2.2}$$

the weak classifier h_n^{weak} gets assigned a weight

$$\alpha_n = \frac{1}{2} \cdot \ln\left(\frac{1-e_n}{e_n}\right). \tag{2.3}$$

Then the weight distribution is updated such that it increases for the samples that are misclassified.

$$w_{n+1,i} = w_{n,i} \cdot \begin{cases} \exp(-\alpha_n) & h_n(\mathbf{x_i}) \neq y_i \\ \exp(\alpha_n) & h_n(\mathbf{x_i}) = y_i \end{cases} \tag{2.4}$$

The corresponding weight is decreased if the sample is classified correctly. Therefore, the algorithm focuses on the difficult examples, i.e. examples that are hard to classify. At each boosting iteration a new weak classifier is added and the process is repeated until a certain stopping condition is met (e.g. a given number of weak classifiers are trained). Finally, a strong classifier $h^{strong}(\mathbf{x})$ is computed as linear combination of a set of N weak classifiers $h_n^{weak}(\mathbf{x})$:

$$h^{strong}(\mathbf{x}) = \text{sign}(conf(\mathbf{x})), \quad conf(\mathbf{x}) = \frac{\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(\mathbf{x})}{\sum_{n=1}^{N} \alpha_n}. \tag{2.5}$$

As $conf(\mathbf{x})$ is bounded by $[-1, 1]$, it can be interpreted as a confidence measure. The higher the absolute value is, the more confident is the detection.
Freund and Schapire [40] proved strong bounds on the training and generalization error of AdaBoost. For the case of binary classification, the training error drops exponentially fast with respect to the number of boosting rounds N, i.e., number of weak classifiers. [156, 160] showed that boosting maximizes the margin and proved that larger margins for the training set are translated to superior upper bounds on the generalization error.

2.2.2 Boosting for feature selection

Feature selection aims to obtain useful features, reduce dimensionality of feature space and remove noisy data. Boosting for feature selection was first introduced by Tieu and Viola [178]. In their work, feature selection from a large set of features is performed by Adaboost. The main idea is that each feature corresponds to a single weak classifier and boosting selects an informative subset from these features.
Training proceeds similar to the described boosting algorithm. Given a set of possible features $\mathcal{F} = \{f_1, ..., f_k\}$ in each iteration step n, the algorithm builds a weak hypothesis

based on the weighted training samples. The best one forms the weak hypothesis h_n^{weak} which corresponds to the selected feature f_n. The weights of the training samples are updated with respect to the error of the chosen hypotheses. Finally, a strong classifier h^{strong} is computed as a weighted linear combination of the weak classifiers, where the weights α_n are estimated according to the errors of h_n^{weak} as described above.

2.2.3 Online boosting learning

Online learning algorithms concern learning each training example once as it arrives and discards it after performing update. An online learning algorithm L_0 takes as its input a current hypothesis h and a new training sample (x, y). After processing, the algorithm returns a newly updated hypothesis that reflects the new sample. The current hypothesis maintains information about training samples it has seen so far. Online learning is necessary when data arrives streaming or the dataset is too large for the batch processing. This property will be exploited in our framework to minimize hand labeling of training samples meanwhile effectively update and improve the classifier.

Given a set of weak classifiers $(h_1, ..., h_n)$ and the corresponding weights $(\alpha_1, ..., \alpha_n)$, a strong hypothesis $h^{strong}(\mathbf{x})$ can be formed. Following the idea proposed by Oza [132], the crucial step of the on-line boosting learning is the estimation of the importance (or the difficulty) of a sample. This can be done by propagating it through a set of weak classifiers. The idea can be thought of as modeling the information gain with respect to the first n classifiers and code it by the importance weight λ (initialized by 1) for doing the update of the $n+1$-th weak classifier. λ_n is either used as a learning rate or by k-times repeated updating $k \sim$Poisson(λ). For updating the waek classifier, any online learning algorithm can be used. The error of the n-th weak classifier is estimated by

$$\hat{e}_n = \frac{\lambda_n^{wrong}}{\lambda_n^{corr} + \lambda_n^{wrong}}, \tag{2.6}$$

where the λ_n^{corr} and λ_n^{wrong} are the sum of important weights of the correctly and incorrectly classified samples seen so far. The corresponding weight is calculated with respect to the error:

$$\alpha_n = \frac{1}{2} \cdot \ln\left(\frac{1 - \hat{e}_n}{\hat{e}_n}\right). \tag{2.7}$$

The importance weight λ for the $(n+1)$-th weak classifier is updated again according

to the error and the output of the weak classifier

$$\lambda_n = \lambda_{n-1} \cdot \begin{cases} \frac{1}{2\cdot(1-\hat{e}_n)} & h_n(\mathbf{x_i}) = y \\ \frac{1}{2\cdot(\hat{e}_n)} & h_n(\mathbf{x_i}) \neq y \end{cases} \tag{2.8}$$

Oza [132] has proved that, if off-line and on-line boosting are given the same training set, then the weak classifiers returned by on-line boosting converges statistically to the one obtained by off-line boosting as the number of iterations $N \rightarrow \infty$. Therefore, for repeated presentation of the training set, on-line boosting and off-line boosting give the same result. In the online boosting algorithm, it is required that the number of weak classifiers is initially fixed.
Note a concerning issue: in off-line AdaBoost, the whole training set is used to update one weak classifier, whereas in the online case one training sample is used to update the entire weak classifiers and the corresponding weights.

2.2.4 Online boosting for feature selection

Boosting for feature selection as described above has been designed to work off-line. Thus, to train a classifier, all training samples must be given in advance. In our work, we use the on-line boosting for feature selection algorithm proposed by Grabner and Bischof [43], which is based on an on-line version of AdaBoost [131, 132]. The algorithm allows to adaptively train the detector and efficiently generate the training set. In the following, we will discuss how the online boosting can be used for feature selection.
Similar to the off-line case, the weak classifiers correspond to the features. As presented in [43], on-line boosting for feature selection is based on the introducing of "selectors" and performing on-line boosting on these selectors (not directly on the weak classifiers). Each selector $h^{sel}(\mathbf{x})$ holds a set of M weak classifiers $\{h_1^{weak}(\mathbf{x}), \ldots, h_M^{weak}(\mathbf{x})\}$ and selects one of them

$$h^{sel}(\mathbf{x}) = h_m^{weak}(\mathbf{x}) \tag{2.9}$$

according to an optimization criterion: we use the estimated error e_i of each weak classifier h_i^{weak} such that $m = \arg\min_i e_i$. Note, that the selector can be interpreted as a classifier as it switches between the weak classifiers. Training a selector means that each weak classifier is updated and the best one with the lowest estimated error is selected. Similar to the off-line case, the weak classifiers correspond to features, i.e. the hypotheses generated by

the weak classifier are based on the responses of the features.
The on-line training version of AdaBoost for feature selection works as follows: First, a fixed set of N selectors, $h_1^{sel}, .., h_N^{sel}$, is initialized randomly with weak classifiers, i.e. features. When a new training sample $(\mathbf{x}, y)$ arrives, the selectors are updated. This update is done with respect to the importance weight λ of the current sample. For updating the weak classifiers, any on-line learning algorithm can be used. The weak classifier with the smallest estimated error is chosen by the selector.

$$m^* = \arg\min_m(e_{n,m}), \quad \text{where} \quad \hat{e}_{n,m} = \frac{\lambda_{n,m}^{wrong}}{\lambda_{n,m}^{corr} + \lambda_{n,m}^{wrong}}. \tag{2.10}$$

The corresponding voting weight α_n and the importance weight λ of the sample are updated and passed to the next selector h_{n+1}^{sel}. The weight is increased if the example is misclassified by the current selector and is decreased otherwise. Finally, a strong classifier is obtained by linear combination of N selectors.

$$h^{strong}(\mathbf{x}) = \text{sign}\Big(\sum_{n=1}^{N} \alpha_n \cdot h_n^{sel}(\mathbf{x})\Big). \tag{2.11}$$

In contrast to the off-line version, a classifier is available at any time and can be directly evaluated, which allows to provide immediate user feedback at any stage of the training process. The pseudo code of the of the Online AdaBoost for feature selection is given in Algorithm 1.

2.2.5 Image Representation and Features

Integral Image Representation

For efficient computation of features (see the next paragraph), integral image representation [187] and integral histograms [143] are used as data structures. An integral image, denoted as II, sums up all the pixel values from the upper left up to the current position. In particular, it is defined on an image I as

$$II(x, y) = \sum_{x'=1}^{x} \sum_{y'=1}^{y} I(x', y'). \tag{2.12}$$

The pre-calculation of an integral image for all pixels can be efficiently implemented in one pass over the image. Afterwards, any sum of any rectangular region can be computed

Algorithm 1 On-line Boosting for Feature Selection (Grabner and Bischof [43])

Require: training example $\langle \mathbf{x}, y \rangle$, $y \in \{-1, +1\}$ (initialized with 1)

initialize the importance weight $\lambda = 1$
// for all selectors
for $n = 1, 2, .., N$ **do**

  // update the selector h_n^{sel}
  for $m = 1, 2, .., M$ **do**

    // update each weak classifier
    $h_{n,m}^{weak} = \text{update}(h_{n,m}^{weak}, \langle \mathbf{x}, y \rangle, \lambda)$

    // estimate errors
    if $h_{n,m}^{weak}(\mathbf{x}) = y$ **then**
      $\lambda_{n,m}^{corr} = \lambda_{n,m}^{corr} + \lambda$
    else
      $\lambda_{n,m}^{wrong} = \lambda_{n,m}^{wrong} + \lambda$
    end if
    $e_{n,m} = \frac{\lambda_{n,m}^{wrong}}{\lambda_{n,m}^{corr} + \lambda_{n,m}^{wrong}}$
  end for

  // choose weak classifier with the lowest error
  $m^+ = \arg\min_m(e_{n,m})$
  $e_n = e_{n,m^+}$; $h_n^{sel} = h_{n,m^+}^{weak}$
  if $e_n = 0$ or $e_n > \frac{1}{2}$ **then**
    exit
  end if

  // calculate voting weight
  $\alpha_n = \frac{1}{2} \cdot \ln\left(\frac{1-e_n}{e_n}\right)$

  // update importance weight
  if $h_n^{sel}(\mathbf{x}) = y$ **then**
    $\lambda = \lambda \cdot \frac{1}{2 \cdot (1-e_n)}$
  else
    $\lambda = \lambda \cdot \frac{1}{2 \cdot e_n}$
  end if

  // replace worst weak classifier with a new one
  $m^- = \arg\max_m(e_{n,m})$
  $\lambda_{n,m^-}^{corr} = 1$; $\lambda_{n,m^-}^{wrong} = 1$;
  get new h_{n,m^-}^{weak}

end for

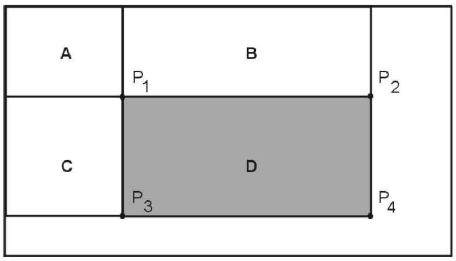


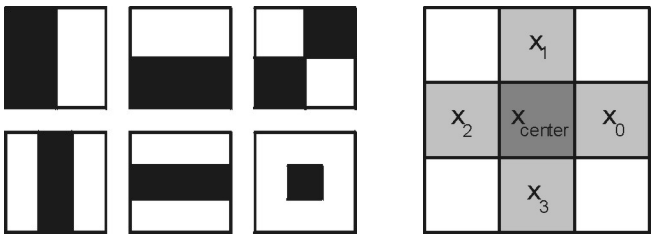
Figure 2.2: Efficient calculation of the sum over a rectangular area. The value of the integral image at Position P_1 is the sum of the pixel values in region A. P_2 corresponds $A + B$, P_3 to $A + C$ and P_4 to $A + B + C + D$. Therefore, the sum over the area D can be calculated by $P_4 + P_1 - P_2 - P_3$.

by only 4 memory accesses and 3 additions, see Figure 2.2 for an example. This allows to do exhaustive template matching when scanning the whole image. The idea can be easily adapted to represent histograms: for each bin one integral image is built separately. These features have been computed over the "norm" rectangular regions. To fit to the object models appearance, the generic angles is needed. This can be done by computing the integral structures at generic angles, i.e by rotated features [94] or rotated classifier [7].

Image Features

The main purpose of using features instead of raw pixel values as input to a learning algorithm is to reduce the intra-class variability while increasing the extra-class variability and adding insensitivity to certain image variations (e.g illumination). In this work, we use three different types of features:

- Haar-like features [187]: The feature value is calculated as the sum of pixel values within rectangular regions which are either positive or negative weighted. These features were introduced by Viola and Jones for face detection and are widely used in computer vision. We use four different prototypes of features, see Fig. 2.3(a). A two-rectangle feature consists of two regions which have the same size and shape and are horizontally or vertically adjacent. For a three-rectangle feature, the sum for the two outside rectangles is subtracted from the sum in the center rectangle. For the four-rectangle feature the difference between diagonal pairs of rectangles is computed. Finally, for a center-feature the center region is subtracted from the surrounding pixels. These features are calculated at different scales.



(a) Haar-like (b) LBP pattern

Figure 2.3: Basic image features used. (a) The value of the Haar-like feature is the difference of the pixel values between the white and the black marked region. (b) Simple version to obtain a local binary pattern value (LBP).

- Orientation histograms [28, 91]: First, a gradient image is computed using the Sobel-filter. A magnitude weighted histogram over the gradient directions is built to represent the underlying rectangular patch. In particular, we use a 8 bin orientation histogram with constant bin size. The basic idea is to describe the appearance of an object part by the gradient informations similar to the famous SIFT descriptor by Lowe [98].

- A simplified version of local binary patterns (LBP) [127]: We use a four-neighborhood, i.e. $2^4 = 16$ patterns, as a 16 bin histogram feature similar to [201]. This is a texture descriptor which captures the statistic of normalized pixel values in a local neighborhood. The LBP-value of a 3×3 image patch $\mathbf{x}$ is calculated as follows (see also Fig. 2.3(b)):

$$LBP(\mathbf{x}) = \sum_{i=0}^{3} s(x_i - x_{center}) \cdot 2^i \quad \text{with} \quad s(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases} \tag{2.13}$$

  The final representation is a histogram of the LBP values obtained by shifting the 3×3 patch in the whole image patch.

Note, that the computation of all these feature types can be done very efficiently using integral images and integral histograms.

To obtain a weak classifier h_j^{weak} from a feature j, we model the probability distribution of this feature for positive and negative samples with $f_j(\mathbf{x})$ evaluating this feature on the image $\mathbf{x}$. Following [43] we estimate the probability $P(1|f_j(\mathbf{x}))$ assuming a Gaussian distribution $\mathcal{N}(\mu^+, \sigma^+)$, i.e. we incrementally update μ^+ and σ^+ for positively

labeled samples and $P(-1|f_j(\mathbf{x}))$ by $\mathcal{N}(\mu^-, \sigma^-)$ for negatively labeled samples.

For the classic Haar-like features, we use a Bayesian decision criterion based on the estimated Gaussian probability density function $g(x|\mu, \sigma)$.

$$\begin{aligned} h_j^{weak}(\mathbf{x}) &= \text{sign}(P(1|f_j(\mathbf{x})) - P(-1|f_j(\mathbf{x}))) \\ &\approx \text{sign}(g(f_j(\mathbf{x}|\mu^+, \sigma^+) - g(f_j(\mathbf{x})|\mu^-, \sigma^-)). \end{aligned} \tag{2.14}$$

For the histogram based feature types, i.e. orientation histograms and LBP, we employ a nearest neighbor learning algorithm. The positive and negative samples are modeled by one cluster for each. The weak classifier is given by

$$h_j^{weak}(\mathbf{x}) = \text{sign}(D(f_j(\mathbf{x}), \mathbf{p_j}) - D(f_j(\mathbf{x}), \mathbf{n_j})), \tag{2.15}$$

where D is a distance metric, in our case the Euclidian norm is used, $\mathbf{p_j}$ and $\mathbf{n_j}$ are cluster centers to be learned of positive and negative samples.
The task is to estimate the two Gaussian distributions (means and variances) for each bin (one for positive and one for the negative class). In the online boosting for feature selection framework, this has to be done in an online manner. [43] has presented two simple approaches based on recursive techniques: Kalman-Filtering and the Approximated median, which have been shown to be efficient with similar performance.

Since we know the resolution of the image, search for cars at different scales is not necessary. Yet cars can appear at any orientation. Instead of training the classifier with different orientations we train it at one "*norm*" orientation. The detector can be made rotation invariant by computing the features at different angles. Lienhart in [94] introduced an additional set of rotated Haar-like features, which comprise an enriched set of basic features and can be computed efficiently. [7] proposed to use different types of Haar-like features. A previously trained classifier is converted to work at any angle, so rotated objects can be detected. A real-time version for the rotation invariant Viola-Jones detector has been reported in [194]. A similar technique is employed in our system: the detector is rotated by increments in 10°. For the orientation histogram features, the rotation can be done by shifting the histogram.

2.2.6 Discussion

In the previous sections, we have described a machine learning approach for feature selection for visual object classification. Online boosting learning method has become widely used in computer vision with remarkable performance on object detection/recognition problems.

Online learning processes each training sample at a time, which allows to incrementally train a classifier as training sample arrives. The state of the classifier is reflected by the current hypothesis, which is learned from the samples it has seen so far. The approach have advantages in situation when the data comes continuously and when the data set is too large to feed into the memory. These advanced properties will be exploited in our framework for learning a classifier for detection of cars from aerial images.

The online learning approach allows us to effectively train a classifier without building a training set in advance. This allows us to avoid tedious work of labeling samples and preprocessing of huge aerial image data prior to training.

In online learning, we can make use of available classifier over time for selective labeling samples and greedy improve the performance of the classifier. As we have fully supervised learning algorithm and a human supervisor, we can always provide good examples to the system for learning, avoid labeling redundancy or noisy data, meanwhile obtain diversifying intra-class variations during learning process.

Moreover, in online boosting for feature selection, the strong classifier is built based on simple features (as weak classifiers) with efficient data representation (i.e integral image), which allow fast processing on huge demand of data from large aerial images.

2.3 The Proposed Framework

2.3.1 Online Boosting Based Learning for Car Detection

This section presents our framework of using online boosting with interactive training to efficiently train and improve the detector. The content is mainly based on our publications [121, 122]. As we have discussed, we use boosting and particularly an efficient integral image representation for fast calculation of car's features. The used features are: Haar wavelets, local orientation histograms, and local binary patterns. We use a novel on-line version of Adaboost to train the detector. It performs on-line updating on the ensembles of features during the training process. By on-line training, we can update the classifier as new samples arrive, therefore require no labeled training data in priori meanwhile minimize

the hand labeling effort.

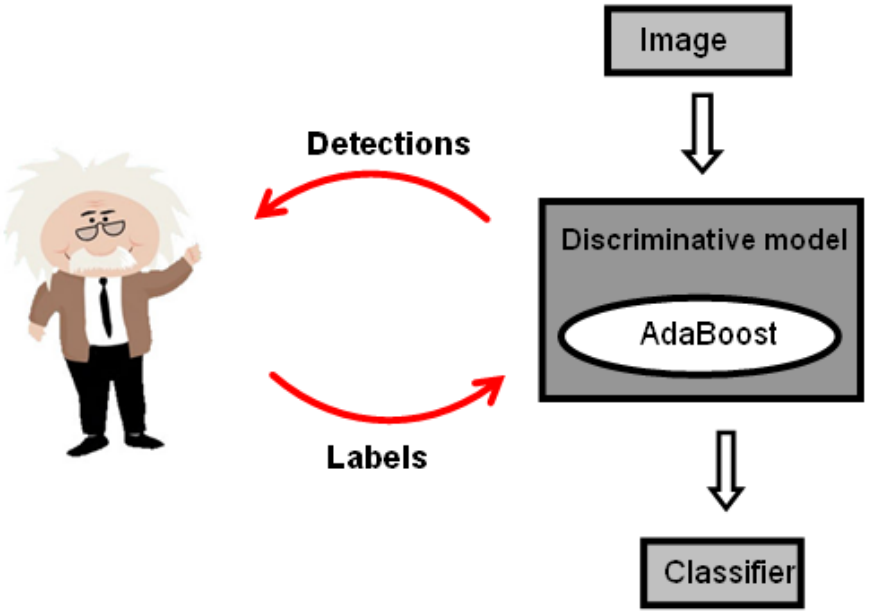


Figure 2.4: The learning process with a human supervisor.

The training process is performed by iteratively labeling samples from the images and updating parameters for the model. The labeled samples can be either positive (containing a car) or negative (a background patch). In order to minimize the hand labeling effort, we apply an active learning strategy. The key idea is that the user has to label only examples which are not correctly classified by the current classifier. In fact, it has been shown by the active learning community [136], that it is more effective to sample at the current estimate of the decision boundary than at the unknown true boundary. This is what we aim at with our approach. We evaluate the current classifier on an image. The human supervisor labels additionally "informative" samples, e.g. marks a wrongly labeled example which can be either a false or missed detection. The classifier is evaluated and updated after each labeling of a sample. The new updated classifier is applied again on the same image or on a new image, and the process continues. This is a fully supervised interactive learning. Figure 2.4 gives an illustration of the learning process with a human supervisor. Figure 2.5 gives an example of training process on a particular image.

The sketch of the on-line training process for car detection as following:

Since labeling of samples in the training phase is an interactive process with human

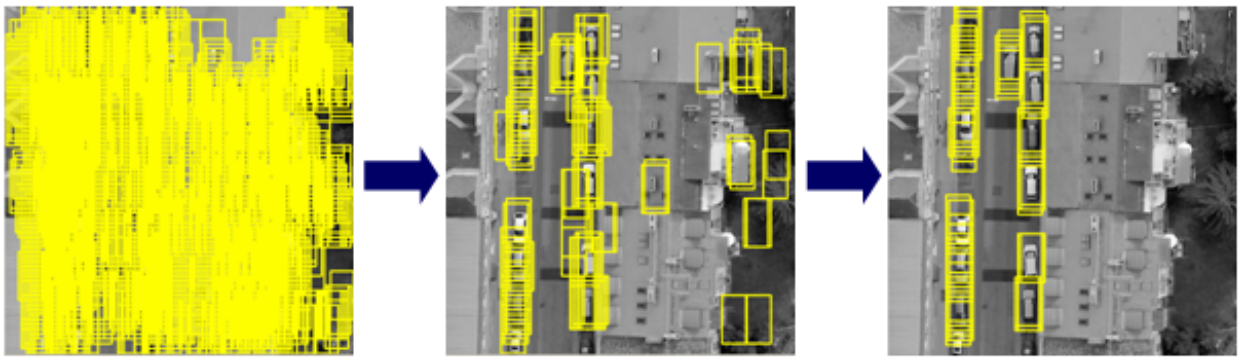


Figure 2.5: Improvement of the classifier over the training process (see also Fig. 2.9 for a postprocessing).

Algorithm 2 On-line training process

```
Initialize parameters for the classifier
while non-stop-criteria do
    Evaluate the current classifier and display results
    Manually label one "good" sample (either positive or negative)
    Update parameters for the classifier
end while
```

supervision, we can intuitively choose to label the most informative and discriminative sample at each update. This allows the parameters of the model to be updated in a greedy manner with respect to minimizing the detection error, meaning that the parameters of the model can be learned very fast. It also avoids labeling redundant samples that do not contribute to the current decision boundary. Therefore this saves a lot of labeling effort. Moreover, by storing parameters of the current training classifier, we can retrain it and make use of pre-trained classifier any time, if necessary.

After training, the detection is performed by applying the trained classifier exhaustively on the images. A car is considered to be detected if the output confidence value of the classifier is above a threshold, i.e. zero. The lower the threshold, the more likely an object is detected as a car, but on the other hand the more likely a false positive occurs. For a higher threshold the false positives decreases at the expense of the detections. The process results in many overlapping detections. Therefore, a post processing stage is needed to refine and combine these outputs. It significantly improves the detection rate. For details of post processing see Section 2.4.3.

2.3.2 On-line Boosting Based Active Learning for Car Detection

In machine learning, active learning refers to a learning method that actively participate in the collection of training examples [177]. Since supervised labeling of data is expensive, active learning aims to reduce the human effort needed to learn an accurate model by selecting only the most informative samples for labeling. The active learning literature offers several approaches for effectively labeling data [25, 161, 177]. In this work, by *Active learning*[*] we mean the learning process itself attempts to select the most informative data for training. This is a process of *selective sampling* [25], in which the selection of good samples for training is partly performed by the learner. In the following, we will give a presentation of our approach.

In a purely on-line learning version as described above, the training process is performed by iteratively labeling samples from the images and updating parameters for the model. The labeled samples can be positive or negative. An active learning strategy is applied. The idea is that the user has to label only examples which are not correctly classified by the current classifier. The classifier is evaluated and updated after each labeling of a sample. By interactive training, one can intuitively choose to label the most informative and discriminative sample at each update, which allows the parameters of the model to be updated in a greedy manner with respect to minimizing the detection error. It also avoids labeling redundant samples that do not contribute to the current decision boundary. Therefore this saves a lot of hand labeling effort.

In this section, we make further steps to greatly reduce the manual labeling effort for training the classifier. Our learning framework is implemented as a wrapper around the online boosting-based learning version. The learning process is shown in Algorithm 3. During the training process, we need to label only few positive samples. The negative samples are automatically generated using the availability of the classifier at each updating iteration. After short time training at the beginning, the classifier is significantly improved, *only weakly detected samples or missed detections are labeled as positive samples, only false positives are used as negatives to update the classifier.* Updates really focus on hard samples. This strategy greatly reduce label complexity and allows faster training. Detailed description as follows:

- Step $2-4$ implement one-class classification learning procedure. In which the clas-

[*]The term *Active learning* applies to a wide range of situations in which a learner is able to exert some control over its source of data. For instance, when fitting a regression function, the learner may itself supply a set of data points at which to measure response values, in the hope of reducing the variance of its estimate, http://themachinelearningforum.org/, Sep. 2009.

Algorithm 3 Online Boosting Based Active Learning

1: Initialize parameters for the classifier as in [43]
2: **while** there exists an undetected (unlabeled) object on current image **do**
3: Label one positive sample
4: Update parameters for the classifier with the labeled sample
5: Evaluate the current classifier on current image
6: **end while**
7: **while** non-stop-criteria **do**
8: Evaluate the current classifier on current image
9: Determine false positives on current image
10: Use false positives as negative samples to update classifier
11: Perform step 2-4 on new image for missed detections
12: Re-update the classifier on seen samples, if necessary
13: **end while**

sifier is able to learn to discriminate object from background solely on the basis of positive samples. This procedure is performed whenever a new training image arrives, and there is no need to perform update if all objects in current image have been well detected.

- The bootstrap procedure is performed at step $7-8$. At each update of the classifier, the current updated classifier is evaluated on current image. This results in a number of detections, which include detected (true) object(s), and usually false positives, e.g. wrong detections. These false positives are patches from background and are actually hard samples to learn, i.e. they are the samples that lie near decision boundary. So, naturally we use these false positives as negatives samples for updating the classifier (bootstrapping). Therefore, there is no need to label negative samples for training.

- The active learning strategy is performed at step $9-10$. After performing few manual marks for labeling positive samples for the updates, the classifier is significantly improved. By evaluating the current classifier on training image, detections are obtained. These detections (on training data) can be used as positive samples to update the classifier, so that the user does not have to label such samples. The human supervisor can decide whether to update the classifier on these true positive samples or not. This is usually not necessary for samples which are well detected (detections with high confidences). We thus force the classifier to update only on hard samples, i.e. weak detection (detected regions with low confidences) or missed detections of either positive or negative classes, with respect to the decision boundary of the current classifier. Therefore fewer update and less hand labeling effort

can be achieved.

For the balancing of training data, an alternative updates on newly generated positive and negative samples can be used. Thus, asymmetric problem is handled naturally by our updating mechanism, without needing a complicated procedure for tuning parameter of the classifier as in [139].

In summary, the training classifier is exploited to automatically generates good samples for learning and incrementally improve itself by update on newly obtained samples. A smooth decision boundary can be obtain since it is refined after updates on really hard samples.

- The verification process: because the classifier is adaptive to newly coming samples, it may make wrong decision on some sample that it has learned so far. The over-adaptiveness may make the classifier unstable. To overcome this, we employ a re-updating strategy on observed samples. This is done by storing labeled samples and regularly reapplying the training classifier to monitor if there is any missed classification so far. If there exists a missed classification on the seen data, the classifier is updated. In the spirit of boosting, the re-update for missed detections can be interpreted as more attention has to pay on hards samples, or to give more weight on samples that is difficult to classify.

By storing parameters of the current training classifier and seen data, we can retrain it and make use of pre-trained classifier any time, if necessary. This results in a more accurate and more stable classifier. Note that, by storing the training samples (obtained during learning process) and allowing the classifier to be re-updated on this seen data, the learning algorithm is no longer "purely online".

After training, the detection is performed by applying the trained classifier exhaustively on the images. An object region is considered to be detected if the output confidence value of the classifier is above a threshold, i.e. zero. This can be done very fast since we use efficient representation of data and simple architecture of the classifier.

2.3.3 Theoretical Justification

By normalizing the boosting weight (see also section **2.2**)

$$\alpha = \frac{\sum_{n=1}^{N} \alpha_n \cdot h_n(\mathbf{x})}{\sum_{n=1}^{N} \alpha_n}, \tag{2.16}$$

it can be interpreted as a confidence measure for the prediction of a sample. Since we are learning a discriminative model, we set our goal to minimize the classification error instead of maximizing the model likelihood. Therefore, instead of learning a full data set, we learn on only a small set of data, which is a set of samples that are close to the estimated decision boundary.

It has been shown in the literature that active learning approaches reduce labeling complexity, achieve high accuracy over random sampling, and reduce generalization error [99, 112, 174]. In our framework, we perform effective sampling by labeling samples at the estimated decision boundary instead of the unknown boundary. It is more likely that the algorithm will make error on samples that close to its current decision boundary. We make a further step for greedy improvement of a detector trained by active learning. Update is performed only on missed detection, which can be either positive or negative. These are samples that lie near the decision boundary and hard to classify. By this update strategy, the algorithm monotonically deceases its true error rate with each mistake and the error rate deceases exponentially with the number of mistakes [112]. Thus, our labeling strategy and updating mechanism greedily reduces error meanwhile progressively improving the classifier.

2.4 Experiments and Performance Evaluation

In this section, we first introduce the data sets of aerial images which we use for the experiments. We employ the Online boosting for feature selection proposed in [43] for the learning framework for car detection as described in section 2.3.1. A summary of training, detection and post processing are described after that. Finally, performance evaluation of the frameworks are presented.

2.4.1 Data Set

The aerial images used in this work consists of images from two large data sets acquired by the UltraCamD camera. The first data set was acquired in Summer 2005 over the inner city of Graz, Austria. The Graz data set consists of 155 images flown in 5 strips. The along-track-overlap of this data set is 80%, the across-track overlap is approximately 60%. The ground sampling distance (GSD) is around 8 cm. The second dataset was acquired in the winter of 2005 capturing the city center of Philadelphia. It consists of 158 images with

an overlap of 90% and a sidelap of approximately 60%. The ground sampling distance is about 10 cm.

In the experiments we use two different datasets of aerial images as described, which include the Graz data set and the Philadelphia data set. With the ground sampling distance is about 8 cm and 10 cm for the Graz and Philadelphia data, respectively, a car is supposed to consist about 24×50 image pixels. Because we know the resolution of the aerial images, we can specify a rectangle, which is a patch at the size of a car. This has to be carefully chosen to cover the area, which contains a car in the middle and a narrow margin (see Fig. 2.6(a)). This is done in order to include some marginal information just surrounding of cars. Usually the car's length is double its width. We have chosen the patch size to be 35×70 pixel.

In this work, only some parts (sub-images) of the large images of the two data sets were employed for the training and testing. From now we will refer to these large scale sub-images as images. For training the system, we used two images of size 4000×4000. One of the two is from Graz city and the other is from Philadelphia. However, not all parts of these images were used. For testing, we used four such images, three from Graz and one image from Philadelphia. Each image also has a size of 4000×4000. The test sets are separated from the training sets. The three images from the test set *Graz* contain 958 cars (324, 288 and 346 respectively from individual image). The test set *Philadelphia* contains 1495 cars. We use gray-valued images obtained from the original multi spectral images for training and testing.

2.4.2 Training

We employ the framework of Online boosting for feature selection proposed by Grabner and Bischof [43] for learning. The system is started with a random classifier which comprises of 500 weak classifiers and 250 selectors. A training image (usually contains some objects to learn) is loaded to the system. The human supervisor starts the training process by labeling one sample, i.e. specifying a bounding box containing the object (car). The system performs update parameters for the classifier with respect to the provided sample. The classifier is improved on-line after the update on the labeled training samples provided by the user. Thus, we make use of the advantages of active learning.

During training we have labeled 1420 samples. There are 410 positive samples, each sample containing a car, and 1010 negative samples, each showing diverse background

(a) (b)

Figure 2.6: Examples of positively (a) and negatively (b) labeled training samples during the on-line training process.

patches (for examples see Fig. 2.6)[†]. The more informative the samples are, the faster the system learns. Moreover, the training samples can be diversified and adjusted during training to capture the variability of the real data. That the number of positive samples is much smaller than the number of negative samples stems from the fact that the variability of the background is much larger than of the cars. In comparison with other object (car) detection systems, our system needs quite a small number of training samples. As can be seen in Fig. 2.7, after several training iterations almost all cars which have a distinct appearance and fit to the (angle of the) detector are detected.

Fig. 2.9, (a) and (c) depicts the detected objects without the refinement step, (b) and (d) are the detected points after applying mean shift clustering. Finally, Fig. 2.8 shows the continuous improvement of the classifier over time, i.e., for an increasing number of labeled training samples, on Graz data set.

Since we want to have good training samples for fast training the classifier, we first train the detector on Graz data. This data set is less noisy due to good imaging quality (in term of sharpness and contrast). We then evaluated the system on both test sets. The performance is quite good on Graz data set. However, it drops significantly on

[†]Since we train on-line a classifier is available all the time. An acceptable result can be obtained after labeling about 800 samples. The longer the training, i.e., the more samples are labeled, the better is the performance.

Philadelphia data. So further training is needed to cope with variability of real data. As we expected, after training with few samples from the Philadelphia data, the classifier adapts quite well and reach the performance as reported in the following section. Note that we keep the same parameter settings as well as the same car patch size for both data sets in training and test phases, except at post processing step. The only parameter needs to be adjusted is a threshold of the confidence. For the Graz data set, distinctive car's features are easily obtained due to good imaging quality, this value is set higher to avoid some false positives that may occur. For the Philadelphia data set, the value is smaller.

2.4.3 Post Processing

Following [42] we use non-parametric clustering-based object detection derived from the probability distribution of classifier output. The strong classifier generates a probabilistic output. For each image location U we obtain multiple outputs P_k representing object appearance's probability (in our case the confidence $conf(\cdot)$ of the strong classifier) at each angle k of the image. To obtain a distribution of object probabilities for each rotation angle, we apply kernel density estimation. Let $\{U_i\}_{i=1,\ldots,n}$ denote the image locations where classification is performed. For each angle k we obtain a probability density estimate

$$\hat{p}_k(\mathbf{u}) = \sum_{i=1}^{n} P_k(U_i) \cdot K_k\left(\frac{\mathbf{u} - U_i}{W}\right), \tag{2.17}$$

where K_k is a two-dimensional Gaussian kernel with a size equivalent to the object size W scaled by the confidence of the classifier output. The cumulative density estimate containing the sum of probabilities over all angles is denoted as $\hat{p}_c(\mathbf{u}) = \sum_k \hat{p}_k(\mathbf{u})$. Mean shift clustering is applied to this density estimate to delineate the appearance of objects. In our case, a simple version is used where K is a two-dimensional flat kernel.

The mean shift algorithm is a nonparametric technique to locate density extrema or modes of a given distribution by an iterative procedure [26]. Starting from a location $\mathbf{u}$ the local mean shift vector represents an offset to $\mathbf{u}'$, the nearest mode along the direction of maximum increase in the underlying density function. The density is estimated within the local neighborhood by kernel density estimation where at a data point $\mathbf{a}$ kernel weights $K(\mathbf{a})$ are combined with weights associated with the data, i.e. with sample weights. In our case sample weights are defined by the values of the density estimate $\hat{p}_c(\mathbf{a})$ at pixel

locations $\mathbf{a}$. The new location vector $\mathbf{u}'$ is obtained by

$$\mathbf{u}' = \frac{\sum_a K(\mathbf{a}-\mathbf{u}) \cdot \hat{p}_c(\mathbf{a}) \cdot \mathbf{a}}{\sum_a K(\mathbf{a}-\mathbf{u}) \cdot \hat{p}_c(\mathbf{a})} \,. \tag{2.18}$$

For a uniform kernel K that we use here, it was shown that fast evaluation of Eq. (2.18) is feasible using integral images [8]. Figure 2.9 illustrates the outputs of the system after post processing using mean shift.

Land Use Classification and Street Layer

In some applications, context information is available and can be used for further improvement of the performance. In aerial images there may exist details of roofs, windows, etc. of buildings that look similar to cars and may therefore lead to false detections. These false detections can be eliminated by using results from other processing stages of the interpretation of aerial images [84, 200]. In this work, a street layer obtaining by land use classification [200] is employed as context. The street layer contains road information, which is used for improving the detection rate.
Land use classification is a two-step process performed on multi spectral digital aerial images. For initial classification, RGB and NIR images are used. A support vector machine [185] is trained for classification on these images . For refined classification, additional height data generated by aerial triangulation and dense matching are used. The classification results are data layers for streets, buildings, trees, low vegetation and water. In the context of car detection we are only interested in the street layer. The street layer is used to mask the possible regions such as road or parking lots, where cars may be located. This helps to reduce the number of false positives considerably.

2.4.4 Evaluation Methods

For object detection problem, the task is to determine the existence of an object in a given image, and specify the location of object in the image. In order to give a fair performance evaluation, we discuss some relevant issues of evaluation criteria.

Ground truth

In our framework, there is no need for any preprocessing of training data since the system is updated on-line when the training sample comes. We also do not have to store any information of training data (only updated parameters of detector are kept), so there is

no need for making of ground truth for the training set. For the test set, ground truth is necessary for evaluating the performance of the system. We proceeded a very simple but efficient procedure to annotate the presence of cars in the images. The original image is splitted into sub-images of certain size. Because we know the location of all sub-images in the original one, the location of any point in any sub-image with respect to the original image is available. Ground truth is done by manually marking one point of a car instance, which represents by the coordinators of center point of the car. Only the coordinate of this one point is stored in the database for testing. This is very simple work and resulted in very small annotated data files. As it has been presented in section 2.4.1, we have manually labeled two data sets for testing. They are the *Graz* and *Philadelphia* data sets, each contains 958 and 1495 cars, respectively.

Quantitative measurement

We apply the trained detector on the whole image to detect cars. With the built ground truth, the detection results can be tested against the ground truth to get the true detected objects. Several criteria for quantitative measurement have been used in literature, such as *overlapping criterion*: a detection is accepted if the overlapping area of the ground truth patch and the detected object patch is above some threshold; or *relative distance criterion*: an ellipse is defined for the displacement between ground truth and detected patch (for the center point in x and y directions). In our work, for a quantitative evaluation, we count as a correct detection if the distance between a detected patch's center and a car's center in ground truth is less than half width of the size of a car (i.e. 10 pixels).

For performance evaluation, we use a common measure for object detection namely recall-precision curves (RPC) as in [2]. This measurement considers the number of false positives with respect to the number of positives in the ground truth and the detected true positives.

$$Precision\ rate = \frac{\#true\ positives}{\#true\ positives + \#false\ positives}, \tag{2.19}$$

$$Recall\ rate = \frac{\#true\ positives}{\#true\ positives + \#false\ negatives}, \tag{2.20}$$

$$F - measure = \frac{2 \cdot Recall\ rate \cdot Precision\ rate}{Recall\ rate + Precision\ rate}. \tag{2.21}$$

The precision rate shows how accurate we are at predicting the positive class. The recall rate tells us how many of the total positives we are able to identify. For the object detection there is always a compromise between the precision and the recall rates. This is evaluated by the F-measure as a harmonic mean. In the plots, we show the recall rate vs. the precision rate.

2.4.5 Evaluation of On-line Boosting Based Learning Framework

In this section, we report the performance of the framework presented in section 2.3.1 for the two data sets Graz and Philadelphia. Fig. 2.10 and 2.12 present the detection results in several subimages. They show complicated backgrounds of urban scenes with many car-like objects. The cars also appear with slightly different view angles, different contrast, lighting condition, etc. Many cars are severely occluded by buildings or trees, dominated by their shadow, or have very low contrast. As one can see, all the cars with distinctive features have been detected. Also almost all difficult cars were found. Some partly occluded cars are detected, some are missed. Some objects that look like cars are reported as cars, but with low confidence values and have been removed at the post processing stage. The system can also deal with slightly different sizes of cars. We have trained it on samples of size of 35×70 pixels. We then applied it for detection of cars on both Graz and Philadelphia datasets with ground sampling distances of approximately 8 cm and 10 cm, respectively. The results show no significant differences.

For quantitative visualization, the RPCs characterizing the performance of our framework for the two datasets with the same parameters setting are given in Fig. 2.11 (lower curves).

In some applications such as the estimation of traffic flow, context information can be given. We use the street layer from land use classification for road verification (see Section 2.4.3). This information improves the detection by eliminating false positives (cf. Fig. 2.12).

For a comparison, besides the regular RPC curves, the RPC curves taking into account context information are also given in Fig. 2.11, upper ones. As expected, the performance of the system gets improved.

Experimental results show that in general the performance of our framework is good and even superior in comparison related work [58, 59, 157, 197, 202]. Moreover, it is a robust and automatic system on large scale aerial images. In terms of the detection rate and especially the efficiency: Due to the lack of public available datasets

for evaluation of the system, a fair comparison is not possible. Additionally, different methods have been employed for evaluation. Some related works even did not provide clearly their performance evaluation, only some intuitive results were shown [58, 162].

Exploiting the redundancy in multiple images

The high overlap of the UltraCamD images results in a high redundancy which can be exploited to improve the car detection at no additional costs. [85]: "One can produce as many images within a flight line as one wishes, with no added costs, and thus increase the traditional forward overlap from 60% to 80% or 90%". The high overlaps produce multiple images for each ground point. This can be exploited to reduce occlusions due to buildings and vegetation, providing superior results. As one can see in Fig. 2.13, cars which are occluded by buildings or trees in one image can become visible and detected in other image of the same flight. For applications such as estimation of transportation flow or terrestrial texture restoration, the use of redundancy is certainly very helpful. Since the establishment of ground truth is tedious for overlapping images, we have not yet systematically evaluated the improvement by using redundancy.

2.4.6 Evaluation of On-line Boosting Based Active Learning Framework

Our main goal here is to demonstrate a further improvement of the new proposed framework presented in section 2.3.2 over the conventional online boosting learning algorithm (section 2.3.1). Thus, in the following we will present the efficiency of learning process by this approach compare to the pure online boosting learning, and the performance of the system. For each experiment, we perform training two classifiers on training data with two training approaches. One classifier is trained by pure online boosting mechanism and the other is trained by new proposed strategy, on the same data set. We kept the same parameter setting for the classifier for each experiment.

To train the classifier by this approach, during the training process we have manually labeled 150 positive samples and the system generated 850 negative samples for self-learning; for the classifier trained by pure online boosting method, as reported in Section 2.4.5, the system needed 410 positive samples and 1010 negative samples. As one can see, the new proposed learning strategy requires quite small amount of hand labeling samples for training in comparison to the pure online learning fashion. Moreover, with the more greedy updating strategy, the system can learn faster.

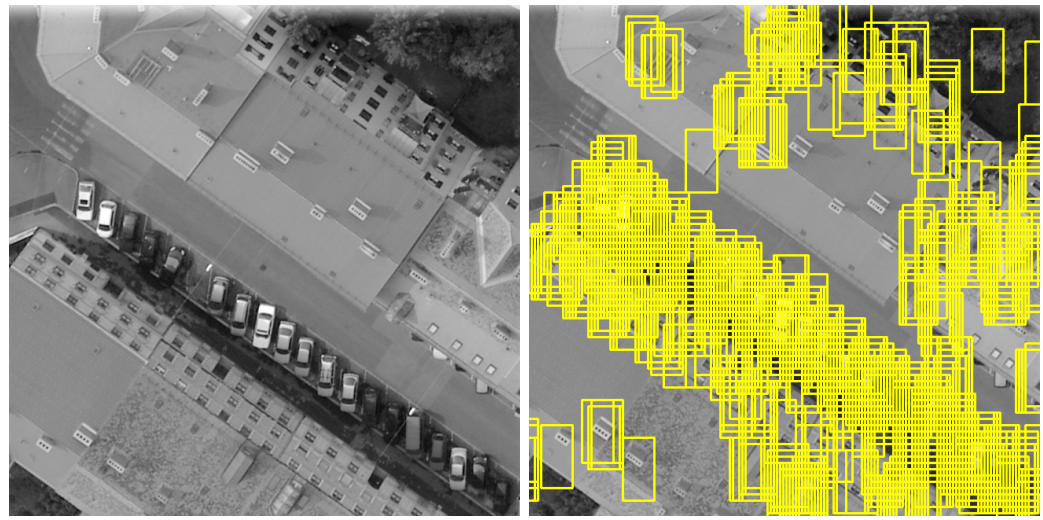
For a quantitative evaluation of performance, we also report the results for the experiment in term of recall-precision curves (RPC). The RPCs characterizing the performance of our frameworks for the two experiments/systems are given in Figure 2.14.

As one can see, the RPC curves of the systems learned by this approach (the upper ones) have some performance gain over the system trained by traditional online boosting approach. This is significant gain over the whole system since with the later proposed approach we used much less hand labeling effort to provide samples to the system for learning. This greatly reduces label complexity to train a reliable object detection system.

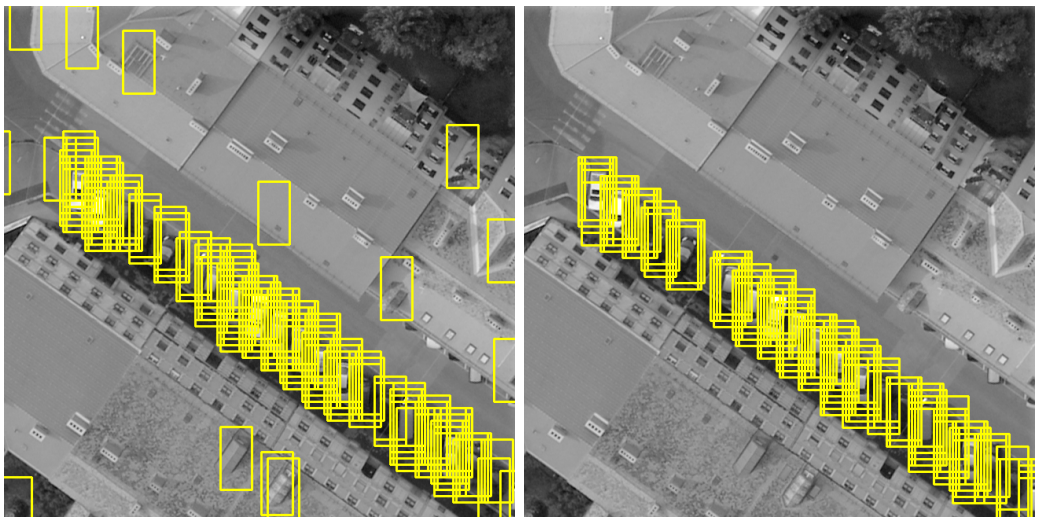
2.5 Conclusion

In this chapter, we have briefly reviewed state-of-the-art related work and pointed out the need to build an qualified system for efficient detection of cars from large scale aerial imagery. We then summarized related methods that we based on to build our framework. These methods include: Boosting algorithm (Adaboost), Boosting for feature selection, On-line boosting learning, and On-line boosting for feature selection algorithms. We have developed efficient frameworks for automatic car detection from aerial images. This is the first proposal to use a state-of-the-art machine learning technique, namely Adaboost, for the detection of cars from large scale aerial images. We have used integral images for efficient representation and computation of car features. Three types of features, Haar-like, orientation histogram and local binary pattern, are employed for generating hypothesis for training the detector. Moreover, a novel on-line version of boosting is used for efficient training of the developed system. On-line learning avoids building huge pre-labeled training set and make use of interactive training. This results in a robust and efficient system for car detection from large aerial images. The system also deals well with the variability of car appearances in complicated backgrounds of urban aerial images. Furthermore, we have proposed to exploit the availability of the classifier during training to automatically generate training samples, aims at reducing hand labeling effort meanwhile keep stable and improving the performance of the classifier.

Experimental results show the applicability and even the superiority of our framework for car detection problem. Possible applications of the framework include estimation of transportation flow, road verification to support land use classification, restoring texture to complete 3D map generation from digital aerial images or 3D city modeling, etc.



(a) (b)



(c) (d)

Figure 2.7: Learning process: Improvement of classifier performance - (a) original subimage from Graz data set, (b) result after training with only one positive sample, (c) after training with 10 samples and (d) final result without post processing after training with 50 samples.

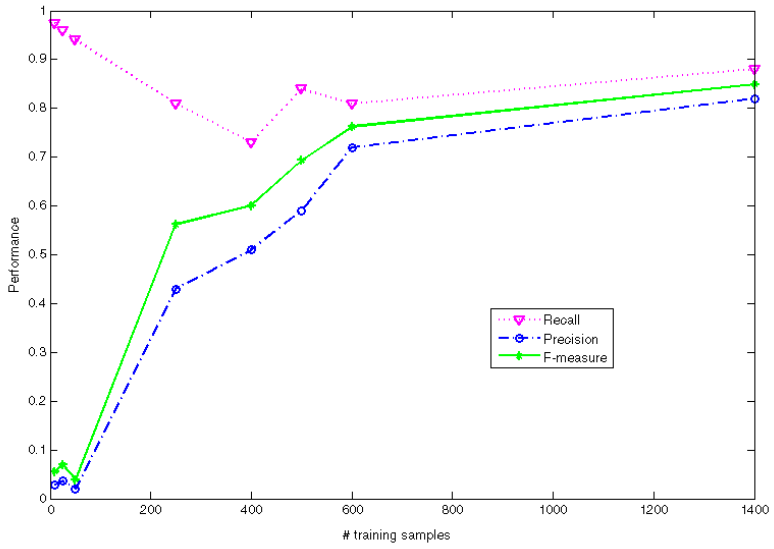
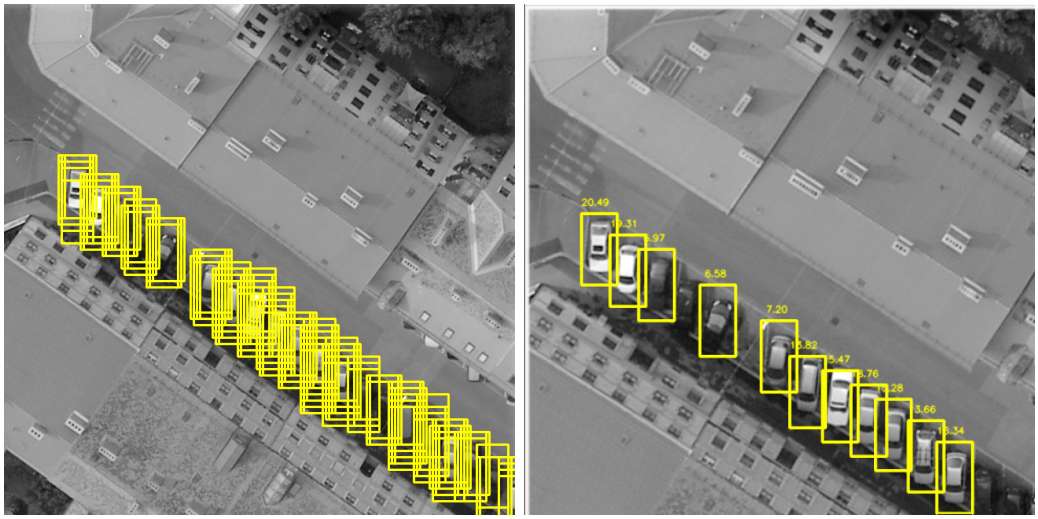


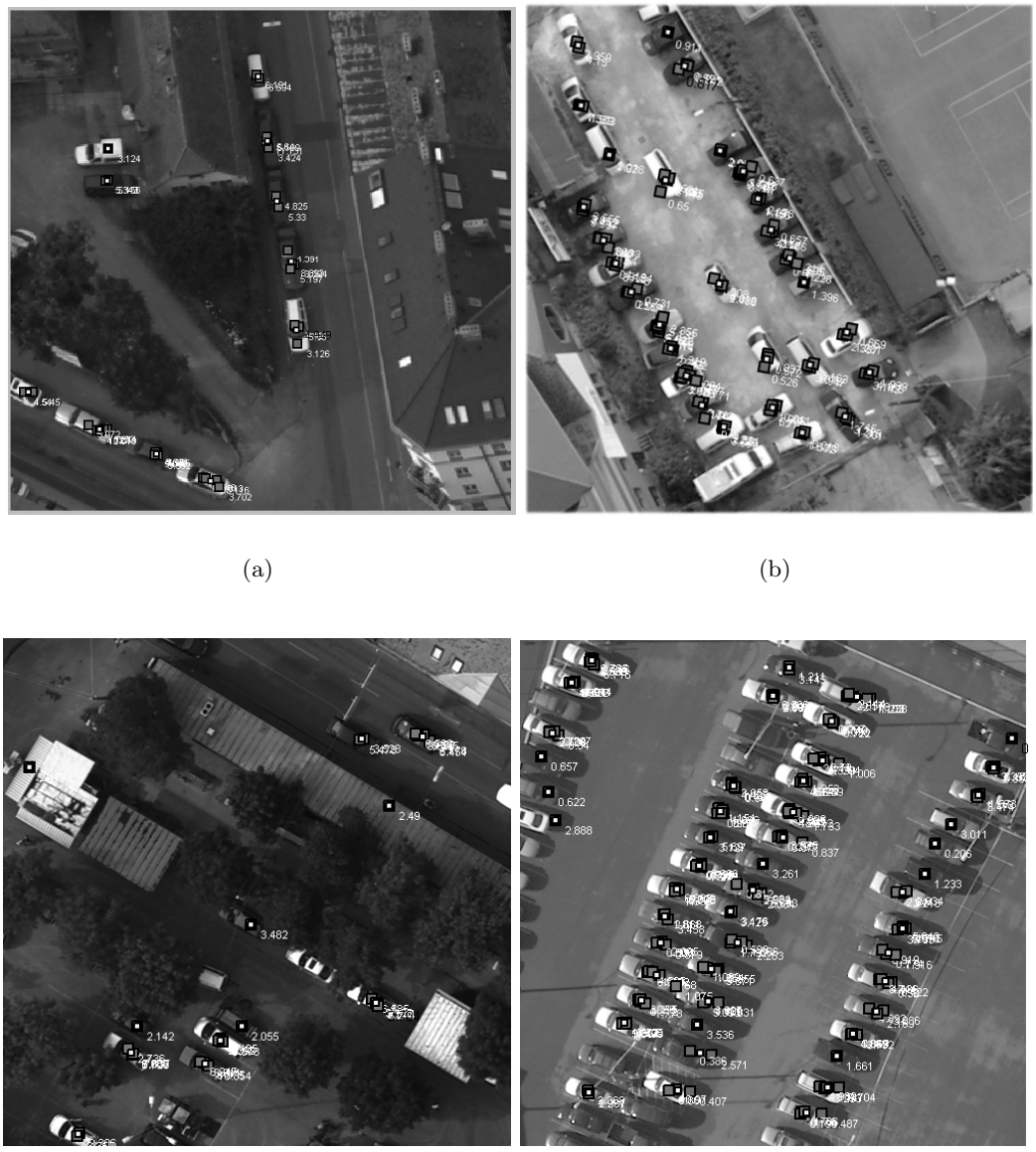
Figure 2.8: Learning process: Performance versus number of training examples, on Graz data.

(a) (b)



(c) (d)

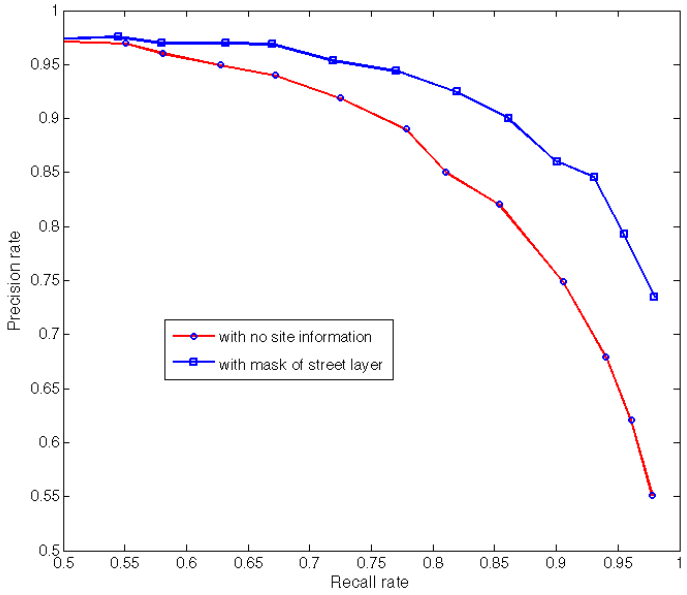
Figure 2.9: Post processing: (a) and (c) are raw outputs of the classifier applied on subimages; (b) and (d) are results after combing multiple detections by mean shift based clustering.



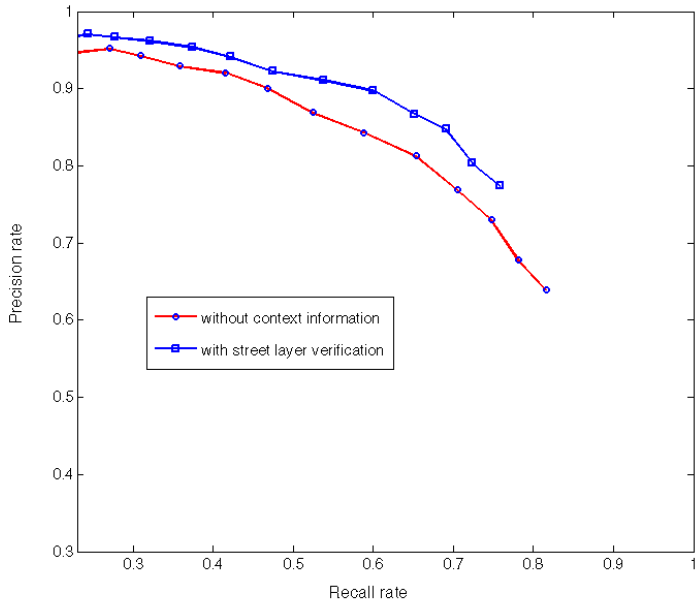
(a) (b)

(c) (d)

Figure 2.10: Results of car detection in large aerial images (left: Graz images, right: Philadelphia images): Cars appear with different orientations and are partly occluded all on highly complicated background. The dark squares represent detections at different angles and bright points are detections after post processing, each point corresponds to one detected car.

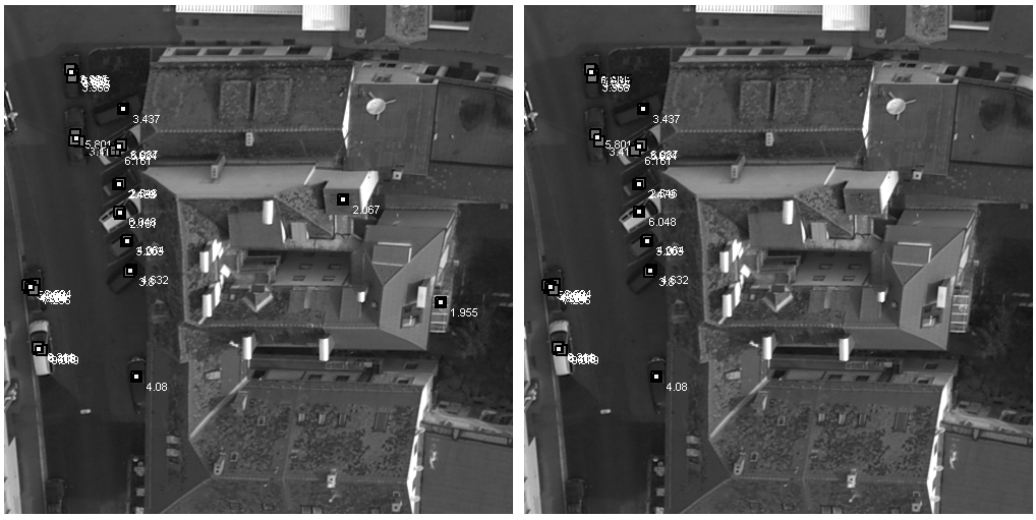


(a)



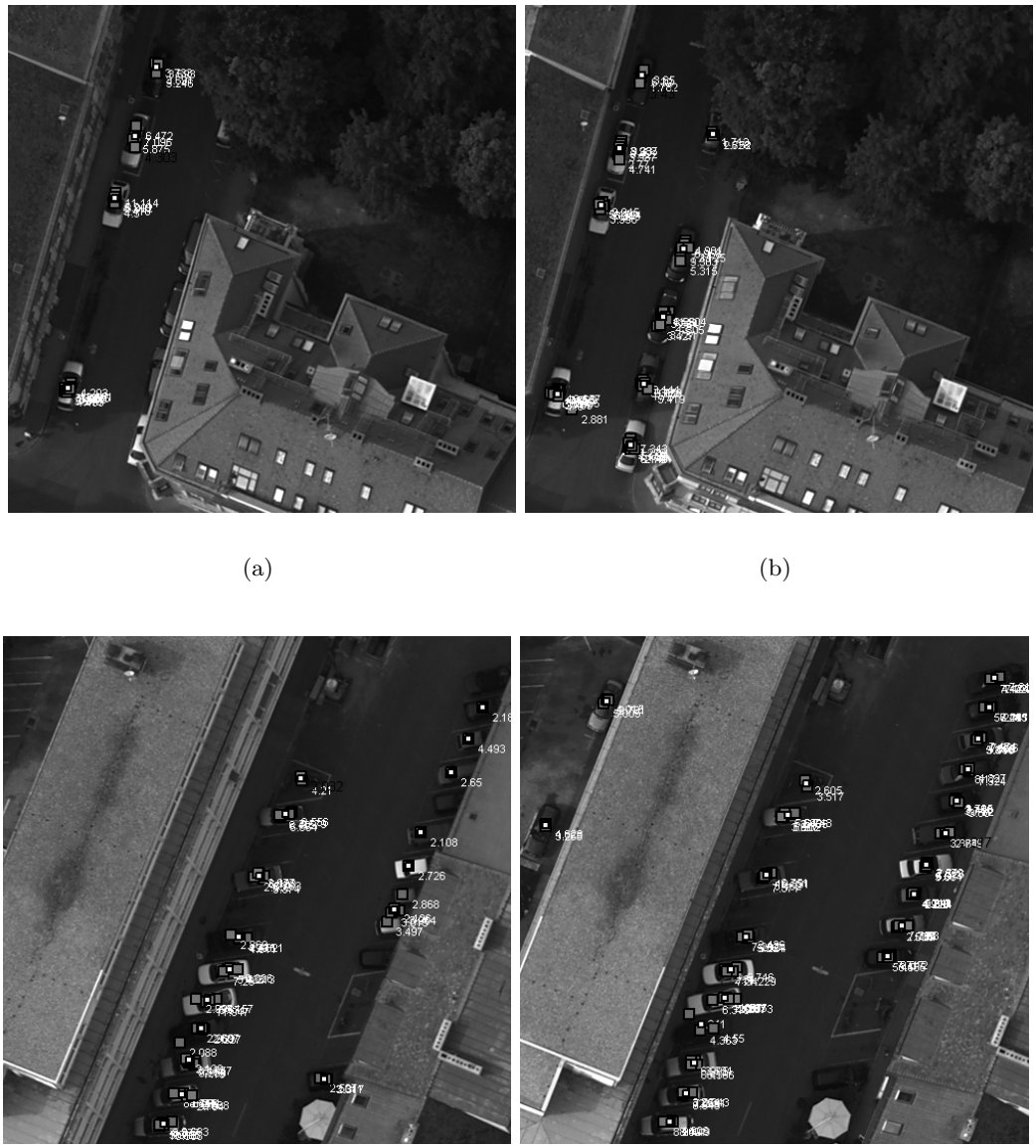
(b)

Figure 2.11: RPC of the system on one image of *Graz* data set (a) and on *Philadelphia* data set (b); Upper curves: Increasing detection performance on the *Graz* and *Philadelphia* datasets when including context information (street layer classification).



(a) (b)

Figure 2.12: Objects on the roof which have been reported as cars are removed using the road mask. The dark squares represent detections at different angles and bright points are detections after postprocessing, each point corresponds to one detected car.



(a) (b)

(c) (d)

Figure 2.13: The utilization of multiple overlapping images with different viewing angles: objects (cars) that are occluded in one image (left images) can be visible and therefore can be detected in another image (right images).

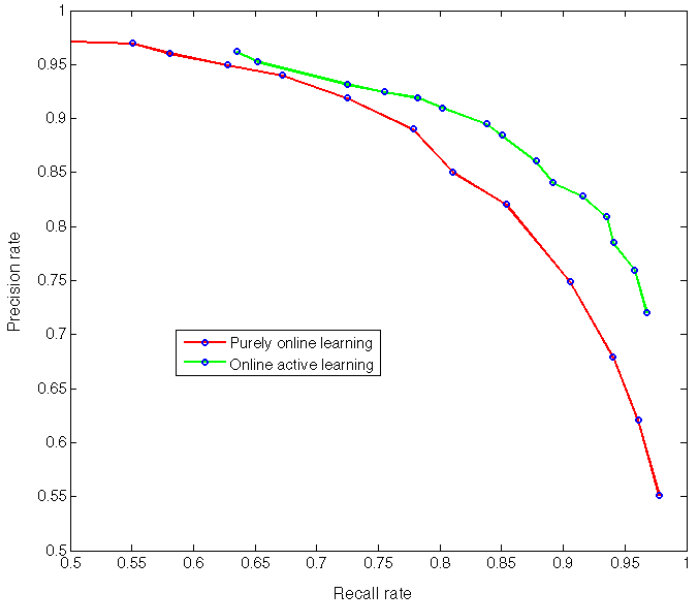


Figure 2.14: RPC of the Online boosting based active system for the Car detection problem.

Chapter 3

HpCRF: Hierarchical pseudo-Conditional Random Field Model and the Building Detection Problem

Contents

The goal of this chapter is to develop a novel hierarchical probabilistic model for the detection and segmentation of complex objects, i.e. buildings from aerial image. The segmentation aims to assign labels to every pixel in an image such that pixels with the same label share certain (visual) properties. This process results in locating the object and its boundaries in images, therefore detection of object.

The content of this chapter is organized as follows: Firstly, we give an introduction and some related work about the buildings detection problem. Secondly, we will discuss the related theoretical issues of probabilistic models that we based on to build our

model. Thirdly, we present our new model, namely Hierarchical pseudo-conditional random field (HpCRF), for the detection and segmentation of buildings from aerial image. We then present our novel algorithm for efficient learning parameters of the proposed model. Fourthly, in the experiments we highlight the results of applying the proposed model for the task of building detection and segmentation from aerial images.

Finally, we end with the conclusion, discussion and the outlook for the future work.

3.1 Introduction to the Building Detection Problem

Automatic detection and modeling of objects, particularly building object, from aerial images is of importance for various practical tasks. Traditional applications are those of cartography and aerial photo interpretation. Recent demanding applications include 3D city modeling, urban planning, Internet applications, virtual site tours, etc. The topic has drawn intensive research in the last decade for automated processing. Besides the practical application needs, building detection and description also provide an excellent domain to explore the problems of image segmentation and scene understanding in computer vision. The problem of building detection is difficult for many reasons. Buildings are complex object with many architectural details. Rooftops are usually composed of different materials with different reflectance properties, which sometimes have low contrast to ground. Building are located in urban scene, that contains various objects from man-made to natural ones. Many of those are in close proximity or disturbing, such as parking lots, vehicle, ground street, lamps, trees. Some objects are occluded or cluttered. Figure 3.1 gives an example of a urban scene of Graz city: buildings appear in complex shapes, may stay in blocks connected to each other with various architectural details; building roofs show variant reflectance, the gray roof tops are very similar to street layer. These difficulties make the problem of building detection challenging.

Among the various practical aspects of the problem, we aim at a concrete task: to detect the appearance of buildings at pixel level. The problem concerns to perform segmentation of buildings or can be understood as building footprints extraction*. This also addresses the task of and semantic modeling, i.e. assigning label for the object class. The detection and segmentation of buildings is necessity for many task, such as *change detection* for map revision (detect new built or demolished houses for updating cadastral maps) [21], or provides building footprints for the next steps of building extraction and

*Building footprints extraction concerns to determine the position and the extent of buildings.

Figure 3.1: An example of complex buildings appearance in aerial image of Graz data set.

reconstruction [64, 138]. Semantic labeling provides knowledge about what the texture describes, i.e. we want to know certain part of the image is a car, a building or a meadow, etc.

With the success of aerial imaging technology, high resolution aerial image can be obtained cost-effectively. Multiple sources of data have become available and make it suitable for the task [47, 85, 200].

Over the years, automated building detection from aerial image has been being an active research topic. There have been a lots of proposed methods for solving the problem of building detection in literature [64, 106]. These approaches are different in the use of data sources, the used models and the evaluation methods [108, 115, 137, 169, 196]. However, how to exploit and integrate multiple sources of data efficiently in a novel learning model, to obtain satisfying performance of the detection and segmentation of buildings at pixel level, is still an open problem.

This chapter aims to develop a new hierarchical probabilistic model for categorical recognition of complex objects class from aerial image. Here, we treat the problem as a binary classification task. A classifier is learned from a set of training data $D = \{(x, y)_i\}, i = 1..M$. The classifier is then applied to map each aerial image pixel x_i

to a bit $y_i \in \{+1; -1\}$, corresponding to either building or non-building.

Related Work

The reader is referred to the Appendix A for general object recognition related work, and to section 3.3 (the Introduction section) for the related work on random field models. In this section we briefly highlight some work directly related to the problem of object (buildings) detection from aerial images.

Buildings detection and extraction is very active research topic in photogrammetry and computer vision at least in the last two decades. The reviews of Mayer H. [108, 110] presents rather comprehensive approaches to the problem in literature. There have been a number of approaches for solving the problem of building detection and reconstruction from high resolution aerial images. They are typically different in type of data and the used methods.

Some works use single intensity image only [96]. Some works use data from multiple aerial images, including color and high field data [64, 115, 196, 206]. Early works mainly used geometric image features for feature extraction. Edges, corners, 2D, 3D structures are used as primitive components to identify buildings. Object size (area, circumference), object form (roundness, compactness) are used as distinguish features; combination detected image edges and detected homogeneous image regions in a hierarchical aggregation process are used to characterize buildings, e. g. Nevatia et al. [120], Fischer et al. [39]. Regrettably, these approaches often fail as soon as the building structures become too complex [35].

In some works, rooftops were mainly used as important cue to indicate the presence of buildings. The features are employed for construction of buildings as well. A perceptual grouping method or a geometric based method is then employed to detect and reconstruct buildings. One benefit of this approach is to allow to perform building detection and 3D building reconstruction at the same time. However, the system is usually complicated due to the requirements of preprocessing phase, and a hierarchy model for comparing and combining building primitives. In many situations, the system failed to find a building or to find a correct description of the building. An interactive system is needed to correct these errors with human user interactions [96, 134].

Regarding the **building change detection**, many related work on are based on methods for land cover classification, in which buildings are firstly detected and then compared to the outdated database in order to detect changes in the scene [107, 171]. Olsen and

Knudsen [128] proposed a two-stage approach for the change detection. First, a trained classifier (from the existing map) is applied on RGB/IR images to segment and classify into building and non-building. A rule-based system is used to achieve a final decision about the change status. An existing building is confirmed if it is verified in at least one image and new buildings are accepted if detected in both RGB and IR aerial images. Champion [20] proposed a two-stage workflow for the change detection process. The first step consists in verifying automatically buildings through a hypothesize-and-verify process: the initial description of the database (extracted geometric primitives) is used to guide the change detection process. The second step consists of extracting new buildings from geometric considerations. The above-ground mask is morphologically compared to the initial building mask (derived from the vector database) and the vegetation mask and new buildings are extracted. The output of the method consists of a change map, in which each building is labeled as unchanged, demolished or new. Matikainen et al. [107] proposed a system for building detection from laser scanning data and aerial colour orthophotos. The DSM is firstly segmented and classified into ground and buildings-or-tree objects. Buildings are then separated from trees through a classification tree based approach. They are later compared to the existing map in order to find out which buildings have been changed, deleted or constructed. Beyond the feasibility to use such a classification-based method in a change building detection process, this paper puts the emphasis on the good performance and the high level of automation of the approach [22]. Rottensteiner [155] proposed a three-stage framework for the change detection. First, a Dempster-Shafer fusion process is carried out on a per-pixel basis and results in a classification of the input data (CIR images and a DSM), then on a per-region basis (after performing regional grouping of building pixels) to eliminate regions corresponding to trees. The actual change detection process is then proceeded, where the detected buildings are compared to the existing map, which results in a detailed change map.
Finally, the two noteworthy real life applications on building change detection and update procedures are the ATOMI project [6, 103] and the WIPKA[†] project, which have been considered as fruit of a productive cooperation between academic research institutions and mapping agencies [22].

Concerning the buildings detection in terrestrial images, Werner and Zisserman [192] presented a geometric approach by making use of the regular structure of buildings, e.g. symmetry or the existence of vanishing points. Dick et al. [31] has present an approach on

[†]http://cmsv021.rrzn.uni-hannover.de/170.html, Jul. 2009.

facade interpretation from terrestrial wide-baseline image sequences. The interpretation of building facades is currently studied by Reznik and Mayer [150]. They make use of Implicit Shape Models combined with plane sweeping to obtain a 3D interpretation of facade planes including the windows from terrestrial image sequences. Drauschke and Förstner [35] presented a method for detecting buildings and building parts using appropriate feature selection.

Some of related works attempted to use contextual information to improve classification result, e.g. intuitively counting building evidences surround a building to decide keeping or rejecting a building hypothesis [115].

Recent approaches have employed graphical models for integrating further information about the content of the whole scene, cf. Kumar and Hebert [75], Verbeek and Triggs [186]. Korc and Forstner [70] employed Markov random field model and shown that state of the art parameter learning methods can be improved and that employing the approach for interpreting terrestrial images of urban scenes is feasible.

There have been some attempts to use conditional random field to model contextual information for detection of urban areas [203] or objects from aerial images [197]. However, non of those works (up to our knowledge) models the inter-feature dependences and spatial context in a consistent, sound principled manner for the problem of building detection and segmentation from aerial images at pixel level of accuracy.

3.2 Background: Random Field Models

In this section we will introduce random field models and briefly discuss related issues that we based on to build our HpCRF model.

We are dealing with the task of classifying image sites (pixels or regions). We first define some notations:
Let the observed data from an input image be $X = \{\mathbf{x}_i\}$, where $\mathbf{x}_i$ is the data from the site i^{th}, S is a set of indexes of all image sites. For image labeling, a site can be a pixel, and a class can be person, car, building, etc. The problem is to find the most likely configuration of labels $Y = \{\mathbf{y}_i\}$, where $\mathbf{y}_i \in \xi$ is set of classes. As we treat the problem of object detection and segmentation as a binary classification, each image pixel with a feature vector $\mathbf{x}_i$ is mapped to a bit $\mathbf{y}_i \in \{-1, +1\}$, corresponding to either object or non-object class *

*We use capital letters X, Y to denote the whole set of image sites and the entire set of labels, P to denote the model distribution, small letters x, y to denote an image site and a particular label, p the

In traditional approaches for image classification problem in computer vision, notable classifiers such as neural network, support vector machine, boosting are usually used. In which, the decision of object class is usually made based on local image data only, e.g. extracted features from a pixel or local image patch. However, naturally the local image sites exhibit dependent properties which can be modeled through different aspects, such as local pixel context, image statistics, scene semantic, etc. Among which, the spatial context constraint is a general and significant one. The spatial dependency should be exploited appropriately to improve the classification performance rather than classify each of image site individually. This has led to a wide range of research on random field models to incorporate contextual potentials into the decision of object classes.

Markov random field (MRF) theory provides a convenient and consistent way to model contextual properties, which can be achieved by characterizing mutual dependencies among entities using conditional random field distributions. Random field models, include Markov random field and Conditional random field, have rooted from Graphical model with solid theoretical background. In the following we will briefly present the sound principles and some applications of those models.

3.2.1 Markov Random Field Model

Markov Random Field (MRF) models allow to take in to account the spatial dependencies in a principled manner. The model and it variations have been used widely for many classification tasks in computer vision [74, 93]. A mathematical definition for modeling local dependency in MRFs is as following [93]:
Definition 1: A set of random variables $Y = \{y_i\}$ *is called a Markov Random Field on a set* S *with respect to a neighborhood* N*, if and only if the following two conditions are satisfied:*

1. $P(Y) > 0$
2. $P(y_i|y_{S-\{i\}}) = P(y_i|y_{N_i})$,

where $S - \{i\}$ *denotes the set difference,* $y_{S-\{i\}}$ *denotes random variables in* $S - \{i\}$*, and* y_{N_i} *denotes the neighboring random variables of random variable* y_i*.*

Condition 1 is called Positivity, which allows the joint probability of any random field to be uniquely determined by its local conditional probability. *Condition* 2 is the Markovian property. It states that the conditional distribution of an instance y_i is dependent

probability of a random field (or a particular likelihood). Otherwise, the notations will be stated clearly.

only on its neighbors.

Markov Random Fields is traditionally a generative approach, which seeks to maximize the joint probability $P(Y^*)$. The posterior over the labels given the observations is formulated using Bayes rule as:

$$P(Y|X) \propto P(Y)P(Y,X) = P(Y)\prod_{i\in S} P(\mathbf{x}_i \mid \mathbf{y}_i). \tag{3.1}$$

Because of the equivalence of MRF and Gibbs distribution [12, 51], the prior is factorized over cliques defined in the neighborhood of Graph G as:

$$P(Y) = \frac{exp(\sum_{c\in C} V_c(Y))}{\sum_{Y'\in\Omega} exp(\sum_{c\in C} V_c(Y'))}, \tag{3.2}$$

where $V_c(Y)$ is a potential function of labels for clique $c \in C$, C is a set of cliques in G, and Ω is the space of all possible labellings*. From (3.1) and (3.2), with Z denotes the partition function (normalization), and assuming Gaussian likelihoods, the posterior distribution can be factored as:

$$P(Y|X) = \frac{1}{Z} exp\left(\sum_{i\in S} log(P(\mathbf{x}_i \mid \mathbf{y}_i) + \sum_{c\in C} V_c(Y_c)\right). \tag{3.3}$$

This is to make the computation of the model tractable. The Gaussian assumption for $P(X|Y)$ in (3.3) allows straightforward Maximum Likelihood parameter estimation.

A traditional MRF with pairwise label dependencies can be formulated as:

$$\begin{aligned} P(Y|X) &\propto P(Y,X) \\ &= \frac{1}{Z}\prod_{i\in S} A(\mathbf{x}_i,\mathbf{y}_i) \prod_{i,j\in N_i} I(\mathbf{y}_i,\mathbf{y}_j), \end{aligned} \tag{3.4}$$

where $A(\mathbf{x}_i,\mathbf{y}_i)$ corresponds to the local potential of $\mathbf{x}_i$ given a class label $\mathbf{y}_i$; $I(\mathbf{y}_i,\mathbf{y}_j)$ is a interaction potential function that encodes the dependencies between labels at site i and its neighbor j, based on the set of pixels in a neighbor N_i of $\mathbf{x}_i$. $Z(X)$ is a partition function. Thus, a MRF makes an assumption of conditional independence of the observed data: it models the spatial dependencies of labels $\mathbf{y}_i$ and $\mathbf{y}_j$, ignoring observations $\mathbf{x}_i$ and

*A clique c is defined as a subset of sites in S, in which every pair of distinct sites are neighbours, except for single-site cliques.

$\mathbf{x}_j$ in the formula.

MRFs have been successfully applied to many problems in computer vision [93]. However, the independent assumption in MRF model is too restrictive for a large number of applications. This is because of, naturally, image data and labels exhibits complex dependencies. At a high level of vision, there exist correlation between object-object (it's likely that there is a mouse if a keyboard is detected), part-object (one could find a face if eye and mouth are detected), object in the scene (it's more likely that a car is in street scene than in an office scene). At the low-level, there exist the dependencies between observations (i.e. pixel, local image data), between observations and labels, between labels and labels. A part from that, long range interaction and inter-feature dependencies could also be taken into account. This has led to research on discriminative field for modeling contextual properties in literature: Conditional Random Field (CRF), which rooted from the early work of Lafferty [79].

3.2.2 Conditional Random Field

Conditional Random Field (CRF) model allows to relax the conditional independence assumption in MRF to incorporate observed data into the formulation. The formal definition of a CRF is as follows [79]:
Definition 2. Let $G = (V, E)$ be a graph such that $Y = (y_i)$, $i \in V$ is indexed by the vertices V of G. Then (X, Y) is said to be a conditional random field if, when conditioned on X, the random variables y_i obey the Markov property with respect to the graph: $P(y_i|X, y_{V-\{i\}}) = P(y_i|X, y_{N_i})$, where $V - \{i\}$ is the set of all nodes in G other than node i, and N_i is the set of neighbors of the node i in G.

A CRF is a discriminative framework, where the conditional model $P(Y|X)$ is constructed from paired observation and label. The CRF is thus a random field globally conditioned on the observation X. The formulation avoids the need to model the observations $P(X)$, allowing the use of arbitrary attributes of the observations without explicitly modeling them.

The conditional distribution $P(Y|X)$ over the labels Y is defined over the nodes, along with the connectivity structure imposed by undirected edges between them. Let C be the set of all cliques in a given CRF. The clique decomposition theorem states that if the conditional density $P(Y|X)$ factorizes according to a graph G, then the features $\phi(X, Y)$ decompose into terms over the maximal cliques $c_1, ...c_n$ of G: $\phi(X, Y) = \{\phi_c(x_c, y_c)\}$, where every $c \in C$ is a clique of the graph and x_c and y_c are the observation and label nodes in

such a clique. Therefore, a CRF factorizes the conditional distribution into a product of clique potentials $\phi_c(x_c, y_c)$*. Intuitively, a potential captures the compatibility among the variables in the clique: the larger the potential value, the more likely the configuration. Using clique potentials, the conditional distribution over the label can be written as:

$$P(Y|X) = \frac{1}{Z(X)} \prod_{c \in C} p_c(\mathbf{x}_i, \mathbf{y}_i), \tag{3.5}$$

where $Z(x) = \sum y \prod_{c \in C} p_c(\mathbf{x}_i, \mathbf{y}_i)$ is the normalizing partition function. The computation of this partition function is exponential in the size of y since it requires summation over all possible configurations of the labels. Hence, exact inference is possible for a limited class of applications of CRF models.

The flexibility of CRF allows the use of arbitrary attributes of the observations and contexts at different levels. These levels of contexts can be exploited to improve the performance of an *i.i.d* classifier in a random field model. At pixel level, local smoothness of pixel labels can be a local context. At higher level, image regions tend to follow certain configurations, i.e. sky is usually above water or buildings. Similarly, for the problem of part-based object detection, geometric relationship among parts of an object can be a context [54, 76]. In some cases, global context may be useful, for example, it is more likely that one can detect a car in a street scene than in an office scene. Moreover, the prediction confidence of the discriminative classifier can also be used as context [182].

From (3.5) we can derive a formulation of CRF as multiple combining of components of conditional distribution that capture statistical information and context at different levels as follow:

$$P(Y|X) = \frac{1}{Z} \prod_{l=1}^{L} p_l(\mathbf{y}_i|X). \tag{3.6}$$

The number of levels L can be chosen depending on the model built for a particular application. The model can be considered as a conditional form of the product-of-experts [57]. The flexibility of CRF formulation allows to incorporate multiple aspects of data from the image, such as: local statistic of image site, neighboring label fields, regional, global or potentials from higher levels of contexts [54, 77, 179, 182]. This advantage property will be exploited in our model. In our work, the model is structured corresponding to $L = 3$. In our model, the first level is a discriminative classifier that based on local

*Clique potentials are functions that map variable configurations to non-negative numbers.

information from image site, the second level intends to model the interaction relationships between neighboring sites, and the third level models the interactions at higher level (will be detailed later). When $L = 2$, which includes local potential and neighborhood relationships, similar to 3.4, we have a CRF formulated as:

$$P(Y|X) = \frac{1}{Z}\prod_{i\in S} A(\mathbf{y}_i, X) \prod_{i,j\in N_i} I(\mathbf{y}_i, \mathbf{y}_j, X), \tag{3.7}$$

where $A(\mathbf{y}_i, X)$ corresponds to the association potential*, which models the dependency between the observations and the class labels; $I(\mathbf{y}_i, \mathbf{y}_j, X)$ is a (pairwise) interaction potential that encodes the dependencies between labels at site i, its neighbor j and the data term X, based on the set of instances in a neighbor N_i; $Z(X)$ is a partition function.
Note that there are two main differences of the CRF in (3.7) compared to the traditional MRF (3.4). That is, the association potential is explicitly dependent on the entire observation X, not only on the local image site $\mathbf{x}_i$, and the interaction potential in MRF is a function of only labels meanwhile in the CRF, it is dependent of both data and labels. The differences play important role in modeling different levels of interactions of observed data and labels in images. The CRF approach has been generalized to many computer vision problem such as image segmentation, image classification, object recognition, etc. [54, 74, 145, 158, 168, 179, 190].
We will give more discussion about the discriminative random field (DRF) model formulation. DRF is a specific type of CRF which has two main extensions which we will exploit in our model. First, the functionality of the unary and pairwise potentials are flexible: they are designed using arbitrary local discriminative classifiers. This allows to use any discriminative classifier (domain-specific) for learning local as well as interaction potentials without restricting. In fact, researcher have successfully used different classifiers for learning potentials such as boosting, support vector machine or random forest. Second, the DRF is defined over 2-D image lattices for modeling data and the labels interaction in a direct neighborhood. This is suitable for our problem as buildings are spread over the scene, the regular concepts of context may not work‡; and moreover, the aerial image are comprised of multiple data modalities. Therefore, we will consider the model of the

*The association potential is also called the unary or local potential.

‡For example, context as interactions between object-object is not suitable as building can locate next to any object in an urban scene; the context such as "sky is above" would not work as buildings can appear at any location in an aerial image.

following form:

$$P(Y|X) = \frac{1}{Z} exp\left(\sum_{i \in S} A(\mathbf{y}_i, X) + \sum_{i \in S} \sum_{j \in N_i} I(\mathbf{y}_i, \mathbf{y}_j, X)\right). \tag{3.8}$$

The form has been shown to be possible to treat different applications from low-level image denoising to high-level contextual object recognition tasks [72].

3.2.3 Potential - Feature Functions

Looking at the two typical forms of random field models (3.4) and (3.7), there are two terms which basically model the potential functions. In which, the first one models the association potential responded by a local classifier. This is the key contribution to the overall performance of the CRF model (the decision of the object class). The second one is the interaction potential, which usually model the contextual dependencies from different aspects of input data and labels, and maybe at multiple levels of interactions.

In the traditional CRF framework, the feature functions are given and fixed. This could be, for instance, real-valued functions taking on the value of a feature for a particular range of values. In the DRF framework [76], the association potential is modeled as a posterior probability of the class labels given the observation, with the parametric form:

$$A(\mathbf{y}_i, X) = log(\delta(y_i \omega^\top h_i(x))), \tag{3.9}$$

where

$$\delta(.) = \frac{1}{1 + e^{-(.)}}, \tag{3.10}$$

and

$$h_i(x) = [1, \phi_1(f_i(x)), ..., \phi_K(f_i(x))], \tag{3.11}$$

where $\phi_k(.)$ are arbitrary non-linear transforms of the feature functions $f_i(.)$ and ω are the parameters to be learned.

For the interaction potential, in the CRF framework, it is also chosen as fixed real-valued feature functions. Different from that, in the DRF framework, the interaction potential is

represented as a pairwise discriminative model of the form:

$$I_{ij}(y_i, y_j, x) = y_i y_j v^\top \mu_{ij}(x), \tag{3.12}$$

where, for a pair of sites (i, j), $\mu_{ij}(x)$ is the pairwise feature vector, and v are the parameters to be learned.
In the DRF framework as well as many variances of CRF proposed later, an arbitrary discriminative classifier is employed for the local class posterior. This gives flexibility to choose suitable classifier for local potential for tasks in specific domains. In the later frameworks, powerful discriminative classifiers, such as support vector machine (SVM) or boosting [76, 166, 168, 182], tend to be used for the association potential. Moreover, the used function is not a designed, fix-valued one. It can be learned from data and can be employed at different levels in the framework. The same spirit for the interaction potential, more flexible mechanisms are utilized to model the dependencies: at different scales and levels [54], confidence map (probabilistic output of classifier), or the dependencies between different feature types can be used as context [101, 182].

3.2.4 Parameters Learning and Inference

Parameters Learning

The parameter estimation problem is to determine the parameters of the model, with regard to maximize the likelihood of the training data. In the MRF framework, the parameters of the prior random field over labels, $P(Y)$ and the parameters of the class generative model, $P(\mathbf{x}_i \mid \mathbf{y}_i)$ are generally assumed to be independent and are learned separately [93].

However, traditional CRF as well as the DRF framework make no such assumption and learn all parameters of the model simultaneously. The similarity in the form of the MRF and CRF models allow most of the techniques used for learning parameters of MRF, with few modifications, can be used for learning parameters of the discriminative models [76, 79].

In the early work by Lafferty [79], the CRF has been proposed for the segmenting and labeling linear sequences. The exact maximum likelihood parameter learning is feasible as the graph contains no loop, which allows computation of partition function using Dynamic programming. A number of efficient techniques have been proposed to learn the parameters of these models, e.g., iterative scaling [79], conjugate gradient, gradient boosting [33, 189].

However, it is not feasible to compute the partition function using using this technique when a graph contains loops.

The problem becomes difficult to exactly maximize the likelihood with respect to the parameters in a general discriminative random field model.

In the following, we will discuss about parameters learning and inference of the CRF model, particularly the DRF form of the model. Assuming that M training samples are available for training the model. Let $\lambda = \{\omega, \nu\}$ be the set of parameters of the model that need to be learned. CRFs learn these weights discriminatively: the weights are estimated so as to maximize the conditional likelihood $p(y|x)$ of the training data. This is in contrast to generative learning, which aims to learn a model of the joint probability $p(y, x)$.

Given M *i.i.d.* labeled training images, the maximum likelihood estimates of the parameters are given by maximizing the log-likelihood:

$$L(\lambda) = \sum_{i=1}^{M} log(P(\mathbf{y}_i \mid \mathbf{x}_i). \tag{3.13}$$

In the standard maximum likelihood approach, learning the parameter of the model involves the evaluation of partition function Z:

$$Z = \sum_{y} exp\left(\sum_{i} A(\mathbf{y}_i, X) + \sum_{i} \sum_{j \in N_i} I(\mathbf{y}_i, \mathbf{y}_j, X)\right). \tag{3.14}$$

In general, the evaluation of the sum over y in Z is a NP-hard problem as it grows exponentially with the number of label configurations in the label space y. To overcome this, one can use a sampling techniques or an approximation of the partition function, e.g mean-field or pseudo likelihood estimation [93].

Maximization of $L(\lambda)$ with respect to the λ can be done using an iterative techniques such as iterative scaling or gradient-based methods. The gradient of the conditional likelihood is computed w.r.t each parameter of the model. Training methods for CRFs include generalized iterative scaling [10, 79], conjugate gradient (CG), limited-memory BFGS, Stochastic Meta-Descent (SMD) [188].

Maximizing the likelihood requires running an inference procedure at each iteration of the optimization, which can be very expensive. An approximation method is to maximize

the pseudo likelihood of the training data, which is the following:

$$p^*(y \mid x, \lambda) \approx \prod_{i \in S} (P(\mathbf{y}_i \mid MB(\mathbf{x}_i), \lambda). \tag{3.15}$$

Here, $MB(y_i)$ is the Markov blanket of variable y_i, which contains the immediate neighbors of y_i in the CRF graph. Thus, the pseudo-likelihood is the product of all the local likelihoods, $p(y_i \mid MB(y_i))$.

According to the pseudo likelihood approach, the parameters in the DRF form are estimated by maximizing the pseudo likelihood instead of the true likelihood:

$$\lambda^* \approx argmax_\lambda \prod_{m=1}^{M} \prod_{i \in S} (P(\mathbf{y}_i^m \mid \mathbf{y}_{N_i}^m, \mathbf{x}^m), \lambda), \tag{3.16}$$

where m indexes over the training images and M is the number of training images.

Computing the pseudo likelihood is much more efficient than computing the original likelihood, because the pseudo-likelihood only requires computing local normalizing functions and avoids computing the global partition function. However, the maximum pseudo-likelihood tends to over estimate the interaction parameters causing poor solution. This can be overcome by an assuming a Gaussian prior over the parameters such that $P(\lambda \mid \tau) = N(\lambda \mid 0, \tau^2 I)$ where I is the identity matrix [72].

The standard maximum-likelihood training for CRFs requires evaluating the partition function $Z(x)$ for each training instance at each iteration, which can be very expensive even for linear chains. Sutton and McCallum [175] proposed to divide the full model into pieces which are trained independently, combining the learned weights from each piece at test time. This is called piecewise training. By using piecewise training, we need to compute only local normalization over small cliques, which for loopy graphs is potentially much more efficient. Piecewise estimation method has been analyzed and found that it performs well when the local features are highly informative [175].

An alternative that has been employed throughout the literature is to train independent classifiers for each factor and use the resulting parameters to form a final global model. Very recently, there have been several proposals training different classifiers for each kind of classification potential then incorporate them at higher levels [81, 101, 168]. The idea will be explored and exploited in our model.

Inference

Given a new test image X, inference aims to find optimal label configuration y over the image sites. Determining the most likely label configuration for model with interaction between sites is an NP-hard problem. For exact inference, the likelihood of all possible label configurations has to be computed, and from that the best configuration could be chosen.

Maximum A Posteriori (MAP) estimation is a widely used technique that is optimal with respect to the zero-one cost function defined as, $C(y, y^*) = 1 - \delta(y - y^*)$, where y is the true label configuration, and $\delta(.)$ is 1 if $y = y^*$ and 0 otherwise. For the binary classification, the MAP estimation can be calculated exactly using the max-flow/min-cut algorithms for an undirected graph if the probability distribution meets certain conditions [69, 73].

Beside the MAP approach, the Maximum Posterior Marginal (MPM) estimation is also widely used. It is is optimal for the sitewise zero-one cost function defined as, $C(y, y^*) = \sum_{i \in S}(1 - \delta(y_i - y_i^*))$, where y_i^* is the true label at the i^th site. The MPM requires computation of marginal over a large number of variables which is NP-hard. A sampling procedure or Belief Propagation can be used to obtain an estimate of the MPM.

These computations are usually intractable for many problems. Therefore, an approximate inference technique is usually used. Various approximate schemes can be seen in literature, such as loopy belief propagation, generalized loopy belief propagation [117, 198], iterated conditional mode (ICM) [13, 74], graph cut [69]. In the framework of DRF, local MAP is obtained by an estimate using the algorithm Iterated Conditional Modes (ICM), proposed by Besag [13]. ICM maximizes the local conditional probabilities iteratively, i.e. $y_i^* = argmax_{y_i} P(y_i|y_{N_i}, X)$. ICM is a local method which gives local maximum of the posterior. It has been shown to be computational efficient and gives reasonably good results [44, 73, 87].

3.2.5 Hierarchical Structure of CRF

Recently, there have been attempts to exploit contexts at different levels in a hierarchy structure of a CRF model [54, 77, 149, 166]. In the followings we briefly discuss the main idea of those modeling architectures.

In general, most of the hierarchical CRF models build the multi-layered structures based on the scales of interactions, i.e. the interactions can be modeled from local pixel level, regional level of interactions to global image property (e.g. for consistent enforce-

ment).

He et. al [54] proposed multi-layer CRFs, which is a product combination of individual models, each provides labeling information from different scaling aspects of the image: a classifier that looks at local image statistics; regional label features that look at local label patterns; and global label features that look at large, coarse label patterns. The model enforced context through the local and global learned features to account for global consistency, which help to improve performance. Kumar and Hebert [77] introduced a two-layer CRF, which models the interactions in image at two different levels. Each layer is a separate conditional field. The first layer models short range interactions among the sites such as label smoothing for pixel-wise labeling, or geometric consistency among parts of an object. The second layer models the long range interactions between groups of sites corresponding to different coherent regions or objects. Thus, this layer can take into account interactions between different objects or regions. The two layers of the hierarchy are coupled with directed links. Top layer is one node, which superimposes objects. A sequential learning approach is used to learn parameters of each layer. The approach of hierarchical support vector random field [166] incorporates SVMs and multiple layers of CRFs to combine local neighborhood and long range dependencies. The multiple layers is deployed by different numbers of parts for different layers and using connectedness between layers. RBF kernel SVM was used and all parameters of the layers are trained jointly.

All the proposed hierarchical CRF have been reported to improve the performance of the traditional single CRF on the same problems to be compared. All these models share a common property in building feature functions on the input data, that is: each local classifier or each single CRF takes into account all relevant extracted information in one feature vector for the class prediction. The context is enforced in multi scales interaction manners.

Another direction of constructing/exploiting multiple contextual information in a CRF model is to decompose input feature space into different feature types, which represent different aspects of image data at very low level. Different feature functions are used to learn each feature types separately and then combine them in a unified CRF model [60, 101, 168]. The 3D LayoutCRF model proposed by Hoiem [60] combines pixel-level and object-level reasoning to detect, segment, and describe the object. The probability distribution for all latent variables conditioned on the image is given by a decomposition of the following components: The part appearance potentials use local image information to detect which part is at pixel i. The part layout potentials encourage neighboring pixels

to be layout consistent, i.e. to have part labels belonging to the same object instance and in the correct relative layout. The instance appearance and instance layout potentials favor part labellings that are consistent with the appearance, position, scale and viewpoint of each object instance. Part appearance and and instance layout are learned separately then combined at refining stage.

In the work of Shotton [168], visual information is decomposed into different cues including shape, texture, color, location and edge cues. These are learned separately and then incorporated in a unified model. A modified method based on piecewise training [175], which involves dividing the CRF model into pieces, each of which is trained independently, is employed to learn parameters of the model.

3.3 HpCRF: A Hierarchical Pseudo-Conditional Random Field Model

Introduction

In this section we focus on developing a novel hierarchical probabilistic classifier for the problem of buildings detection.

As we have discussed, the probelm of object recognition in general and building detection in particular is a challenging task in computer vision. Many approaches have been proposed for modeling and learning visual properties of object classes. Traditional approaches based on local image patches to extract object features. Visual information, such as color, texture, contour, etc. are mixed together in a single feature vector to represent the instance. While this approach has been widely used in the community and gained significant success, it poses several problems: Visual stimuli are processed separately by observer and different visual cues should play different roles in discriminating the object classes [100]. Beside that, the conventional way of concatenating multiple feature types into a single feature vector to feed to the classifier may cause the problem of over-fitting due to redundancy and correlation in the input data [151]. Moreover, standard learning algorithms, such as Naïve Bayes, Logistic regression, support vector machine (SVM) assume that these instances (image sites) are *independent and identically distributed (i.i.d)*. This is inappropriate in many cases, as image pixels are usually dependent: if a pixel labeled as building, it is likely that the neighboring pixel is also labeled as building; non-building pixels tend to be next to other non-building pixels.

As discussed in previous sections, a family of random field models (MRFs, CRFs) have

been proposed to improve the performance of $i.i.d$ classifiers by modeling the contextual potentials [79, 93]. There have been wide research interests in MRF, CRFs and their variants in the computer vision community. Successful results have been reported [9, 54, 74, 77, 81, 146, 168, 179, 181, 193]. However, there are still a number of issues that should be considered: (i) MRF and CRF are mainly based on a local $i.i.d$ classifier, as mentioned before, which inherits the same spirit of using mixed features for both local and interaction potential terms. While it is observed that each type of feature may have its own context to exploit, and, in some case one feature may inhibit the performance of the other. (ii) Many of the proposed systems use single CRF model, which is limited in the interaction, and there is no interaction between different feature types and their spatial context (potentials) at higher level. (iii) Training and inference are time consuming and the model is only applicable for limited problems.

Recently, there have been several attempts to decompose low-level visual properties for learning a coupled conditional random field for object categorization [101]. Some works used probabilistic prediction values as context for learning a CRF model [182] or a graphical model [71]. There are proposals of hierarchy CRF to exploit contexts at multi levels, aim to improve classification accuracy [54, 77, 166]. We adopt some of these ideas in our model to address the mentioned issues.

Visual objects in general, building objects in particular, can be represented by different feature types that capture different aspects of object's properties, such as color, texture. These features are strong-related, spatially and statistically. The idea is to treat different feature types in separated processes and then integrate them at subsequence stage into a unified probabilistic model. This aims to fully exploit the discriminative power of each individual feature type and to leverage the performance by using context at higher level. In this chapter, we propose a new probabilistic framework in the spirit of random field models, named hierarchical pseudo-conditional random field (HpCRF). The model is structured as two-layer multiple discriminative classifiers. The structure of our model is depicted in Figure 3.2. For a clearance, only one node in the middle is shown with full links.

The model is structured as layered multiple CRFs. The first layer of the model includes several CRFs, each CRF is responsible to a certain feature type and modeled as pseudo-CRF for learning the context. This allows to fully exploit the classification potential of each feature type and its own context. The second layer is built on top of features and classification confidences from the first layer. Multiple discriminative classifiers are employed to learn the classification potential of each feature type together with it context

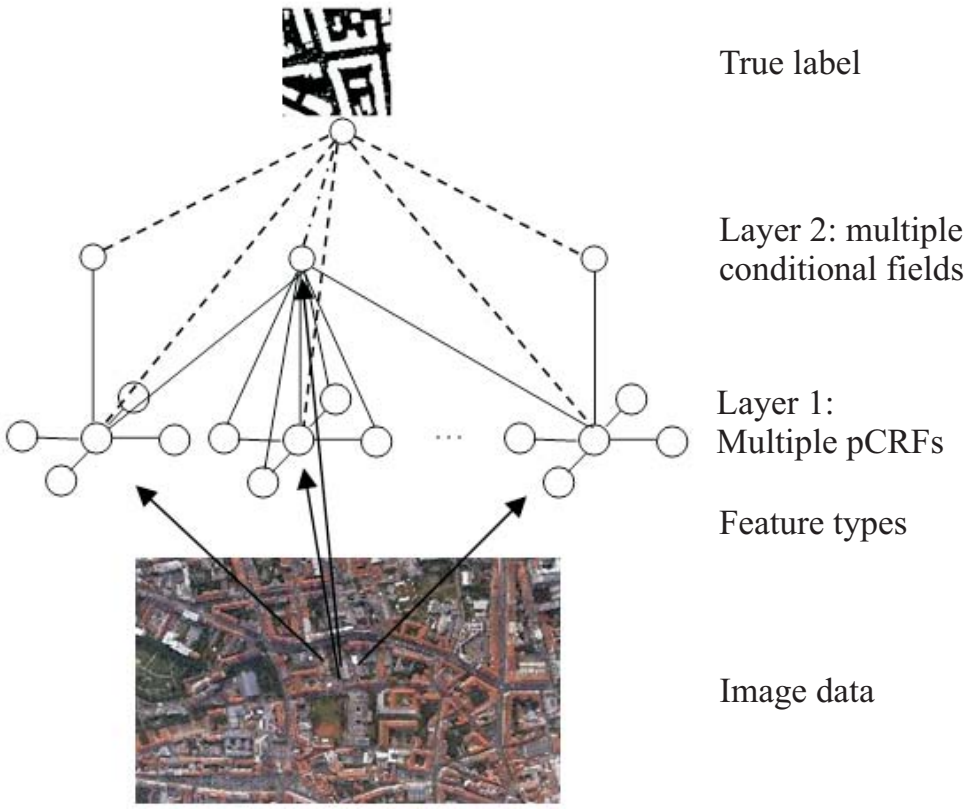


Figure 3.2: The hierarchical pseudo-Conditional random field model, HpCRF.

and potentials from other types. Here we employ the idea of stacked graphical learning: feature vector of certain type is expanded with prediction confidences from its neighboring sites and predictions from other types. This allows to learn inter-feature dependencies and to integrate contextual information efficiently. At the top level, the classification confidences from individual classifiers at the two layers are incorporated and fused to infer the object class. All discriminative classifiers share the same parameter setting and learning mechanism. The proposed system provides a simple yet efficient way to model complex object class by exploiting discriminative potentials from different aspects of image data. Learning and inference are effective, general and straight forward. It can be easily used for a number visual learning tasks.

3.3.1 HpCRF: A Hierarchical Pseudo-Conditional Random Field Model

Multiple conditional fields - Layer 1

A *feature type* includes features which represent the same property of the image data. In fact, different feature types represent different aspects of image data. In other words, feature types are different cues for classification, such as color, texture, contour. For example, texture feature type refers to texture cues, or features which represent a visual pattern that has property of homogeneity; color feature types refers to color cues, or features which represent the color property of the image, etc. With a little abuse, in the following we will use the terms *color feature type, texture feature type, height feature type* to refer to extracted features which represent for color, texture and the height field data information, respectively.

Assuming that the observed data from an image X can be decomposed into T feature types, $X = \{X^t\}$, $t \in T$. Then, the conditional field of the first layer of the HpCRF is modeled as multiple combining of CRFs:

$$P_{(1)}(Y|X) = \frac{1}{Z}\prod_{t=1}^{T} p^t_{(1)}(Y|X^t), \tag{3.17}$$

where each $p^t_{(1)}(.)$ is a CRF defined as in (3.7)*, t denotes a feature type, index *(1)* in $P_{(1)}$ is for the first layer of the hierarchical model.
We make an assumption of pairwise interactions of data and labels on a regular grid. The main reasons of using decomposition of input data and building multiple ensemble CRFs model are: First, as it has been investigated, visual cues such as color, texture. play different roles in distinguishing object classes [101]. These should be treated separately by different processes and combined at later stages to infer the object class [123]. Moreover, since random field modeling approaches try to exploit contextual information to improve detection rate of standard classifiers, it is intuitively sensible that different object's property have their own context where it it more likely to appear. This is especially true in our real world application, where multiple sources of aerial image data are available to use, i.e. the color image and the height data§: it may be claimed that pixels with similar color have similar labels; however, this is not true for the height data: pixels with the same

*Note, the difference in the used data term: in (3.7) X denotes the whole features of the image data, where in the (3.17) X^t denotes one feature type of the image.

§The height field data, which corresponds to the elevation of terrain, is obtained from Digital elevation model (DEM) [200].

height values may belong to buildings or height trees. Thus, the ensemble model comprise of multiple CRFs, each CRF responses to a certain feature type, is to fully exploit the potential of each feature type and its contextual information.

The traditional form of CRF and its variants are computational expensive for learning and inference. Many approximation methods have been proposed to deal with this. Notable techniques are: pseudo-likelihood, piecewise training, contrastive divergence [54, 72, 168, 176]. These approximations only ensure local optimum and affect the performance. Moreover, it is still almost impossible for our application where huge data from aerial images need to be processed. Therefore, we employ the idea of pseudo conditional random field (pCRF), as proposed in [88]. The pCRF is a simplified form of the regular CRF, in which the (local) *i.i.d* classifier is regularized: the discriminative classifier is first trained on the data, which assumed as *i.i.d*. This makes the training efficient. It then relaxes this assumption during inference by including the labels and feature vector from neighboring pixels as a regularized term. Hence, with the pCRF we need only to learn the parameters of the local classifier, i.e. the association potential.

The interaction potential of neighboring pixels is modeled as:

$$I(\mathbf{y}_i, \mathbf{y}_j, X) \quad = \quad d(\mathbf{x}_i, \mathbf{x}_j)\psi(\mathbf{y}_i, \mathbf{y}_j), \tag{3.18}$$

where $d(\cdot)$ measures the similarity of neighboring pixels (a distance function can be used), $\psi(\cdot)$ is the label smoothing factor. Inference of pCRF is performed by incorporating the neighboring potentials into the local potential with the objective to maximize the likelihood: $y^* = argmax_y P(\mathbf{y}|\mathbf{x})$.

Multiple conditional fields - Layer 2

We are interested in a model that can capture the dependencies among different kind of features, labels and contexts. We build the second layer of the HpCRF model based on the features and outputs of classifiers from the first layer. Again, we treat each feature type separately. At this level, we want not only to model the context of each feature type individually, but also to model the dependencies between feature types and their spatial corelations. This enables to exploit the inter-feature dependencies and to learn the interactions between data and labels at higher level. The model at this layer is expressed

as:

$$P_{(2)}(Y|X) = \frac{1}{Z}\prod_{t=1}^{T} p^{t}_{(2)}(Y|X^t, p_{(1)}). \tag{3.19}$$

First, each feature vector of certain type is now expanded with prediction confidences from its related elements. In particular, each original feature vector of certain feature type is augmented with predictions from its neighboring pixels and predictions for it from other feature types. This forms new training sets, which captures the dependencies among local data and neighboring labels of different feature types. We use an aggregate function to build the new training sets: For each feature type X^t, each instance $\mathbf{x}_i^t$ is augmented with prediction confidences from its neighbor and from other feature types, $\mathbf{y}^t_{N_i}$ and $\mathbf{y}_i^{T\setminus t}$, respectively.

$$\mathbf{x}^t_{i,new} = (\mathbf{x}^t_i, y^t_{N_i}, y_i^{T\setminus t}), t \in T. \tag{3.20}$$

Second, multiple discriminative classifiers are then employed to learn on these new training sets. The learning process is performed similar to the learning process at the first layer.

Finally, at the top level, probabilistic values of classifier's outputs at the two levels are fused together to infer the object class. Note that we fuse the classification potentials, not the combination or the hard predictions of the classifiers [57, 123]. The idea of fusing class potentials is similar to cue integration via accumulation. That is even when most of the classifiers provide a wrong answer, the final classifier (the whole model) still has a chance to produce correct result, due to the accumulation effect [123].

3.3.2 Learning and Inference of the HpCRF

Parameters learning

Since the HpCRF model comprised of multiple conditional random fields (CRFs) in a two-layer structure, learning the parameters of the HpCRF includes learning the individual CRF at each layer. As the CRFs at the second layer are based on the output of the CRFs at the first layer, learning has to perform sequentially for the first layer and then for the second layer.

Learning of the HpCRF model is more efficient than other hierarchical random field models due to the pseudo design of the individual CRFs. For each CRF we only need

Algorithm 4 Learning and Inference of the HpCRF model

1: Given training set $D = \{(X, Y)\}$ and a discriminative classifier A; D is a decomposition of T feature types: $D^t = \{(X^t, Y)\}$, $t \in T$.
2: Learning algorithm: For each feature type $t \in T$
3: - Learn the local model: $\mathbf{f}_0^t = A_1(D^t)$
4: - Infer the probabilistic class label for *layer* 1 using pCRF: $\mathbf{f}_1^t = pCRF(\mathbf{f}_0^t, D^t)$
5: - Construct an extended data set: $D_{new}^t = \{\mathbf{x}_{i,new}^t\}$, using Eq. (3.20)
6: - Learn the probabilistic model at *layer* 2: $\mathbf{f}_2^t = A_2(D_{new}^t)$
7: Inference: given new image $\mathbf{X}$, for each feature type $t \in T$
8: - Compute the local classification potential: $\mathbf{y}_0^t = \mathbf{f}_0^t(\mathbf{X})$
9: - Compute the prediction confidence using pCRF (layer 1): $\mathbf{y}_1^t = \mathbf{f}_1^t(\mathbf{X}, \mathbf{y}_0^t)$
10: - Compute prediction confidence (layer 2): $\mathbf{y}_2^t = \mathbf{f}_2^t(\mathbf{X}, \mathbf{y}_1^t, \mathbf{y}_1^{T\setminus t})$
11: Fusion of all prediction potentials to infer class label: $y_i^* = argmax_{y_i} \prod_l P_l(\mathbf{y}_i | y_{N_i}, \mathbf{X}), l = 1..2.$

to learn the parameters for the association potential. In principle, any discriminative classifier which gives probabilistic output for the class label can be used to learn this potential. In this work, we employ probabilistic support vector machine (SVM) [24] with linear kernel as the base classifier. SVMs are classifiers that have appealing theoretical properties and have shown impressive performance in variety of classification tasks. The linear kernel is utilized for its computational efficiency and giving satisfying results*.

The SVM learns the local classification potential for each individual feature type. After training SVMs, we employ the idea of pseudo-CRF [88] (see also previous section) to compute the interaction potential between data and label in a defined neighborhood structure. In this work, we use pairwise interactions on a regular grid.

Pseudo CRF has been shown to be simple and fast. It can give satisfied performance, which is suitable for processing on huge aerial image data.
After obtaining probabilistic output of classifiers at the first layer, new training sets are built as equation (3.20). Again probabilistic SVMs are employed for training these new expanded data sets. We keep the same parameters for learning the individual classifiers, e.g. the number of support vectors and the kernel of the SVM. Therefore, there is no need for hyper parameter tuning. After learning, the learned classifiers are applied on each test image. We then combine the classification potentials (rather than classifiers) to infer the object class.

*In an ongoing work, we have tested the system with the state-of-the-art random forest classifier [167] for learning the association potential. Experimentally, we obtain competitive results for the framework compared to the one used SVM

Inference

In our hierarchical HpCRF model, the inference has to be done for the pseudo CRFs. Because of the pCRF learning using a pseudo strategy, the inference for each feature type is the process of incorporating the regularization term based on the neighborhood relationship.

In general, inference involves inferring true label for each image site (image pixel in our case), which is done by computing the most conditional likelihood: $y^* = argmax_y P(\mathbf{y}|\mathbf{x})$, given feature vector $\mathbf{x}$ and the learned potential functions. Exact computation of this is expensive and undesirable large-scale images data. Approximation methods, such as Iterative conditional mode [13], Loopy belief propagation (LBP) or Graph-cut have been widely used [74, 88, 101, 168].

We want to have a fast inference method for processing on large-scale aerial images. Although there have been many approximation algorithms designed to find the optimal y^*, we will focus on the local method called Iterated Conditional Modes (ICM) [13]. ICM is employed for its fast convergence and produce sufficient accurate results. In our model, we need only to find maximum conditional probability in a local neighborhood of each pixel, which is in the following form:

$$y_i^* = argmax_{y_i \in \xi} P(\mathbf{y}_i|y_{N_i}, \mathbf{x}_i). \tag{3.21}$$

Note that, except at the final node (the top level), where the inference is to find the true label, the inference at the intermediate steps is to get the maximum probability for the class label to use in the next step inside the grand model. The procedures for learning and inference of the model is given in Algorithm 4.

Concerning the complexity, our HpCRF model comprised of multiple pCRFs, for each pCRF the main learning task is to learn the parameters of the discriminative classifier, and not really learn the parameters for the spatial correlation. Therefore, the complexity of the HpCRF just depends on the number of feature types, which are three, and the complexity of the local classifier for learning the association potential, which is linear kernel SVM in our framework.

3.4 Experiment and Result

In this section we will evaluate the proposed framework on a real problem with huge demand of data. We apply the model to the problem of detection and segmentation of

buildings from aerial images at the pixel level. We implement the HpCRF as described in previous sections. We then compare the performance of our HpCRF with several state-of-the-art traditional methods.

3.4.1 Data Set

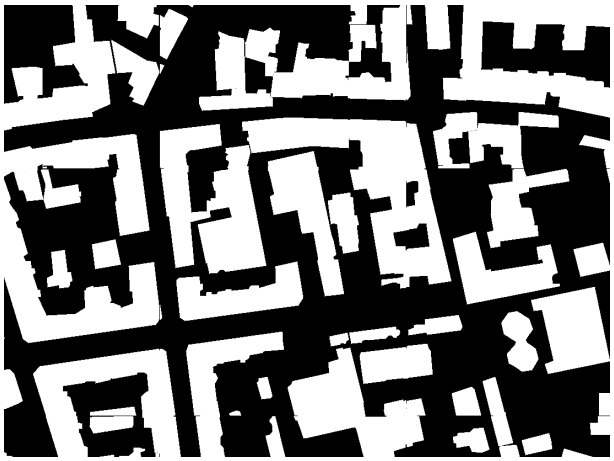
The data set is derived from high resolution aerial images, i.e. the ones produced by the *UltraCamD* from Microsoft Photogrammetry as described in Chapter **1**, section **1.1**, and in Chapter **2**, section **2.4.1**. In this work, we used two types of information from aerial image data, which are the color image and the height field data. Each color image includes three image planes of the three color channels: red (R), green (G) and blue (B). Since the aerial images have been taken with high overlap, a dense matching approach [67, 200] results in the *range images* representing the Digital surface model (DSM). For which we can obtain the relative elevation per pixel from ground, that represents 3D height information. Figure 3.3 shows a typical scene, including the color image, the hand labeled buildings mask and the corresponding relative 3D height data. Such triplets of images are used for the training and the evaluation procedures. For training and testing the model, twelve triplets extracted from huge images are used. Each of these sub-images has a size of 2000×2000 pixels (this is approximately 480 images of size 256x384). The images cover large dense urban areas, which contain various complex objects, such as buildings of variant sizes and complex architectures, road net, parking lots, trees, shadow, water surface, etc.

In the experiments, The training set is separated from the test set. The large images are downscaled 1/4 for reducing computational burden. Downscaling is reasonable as in large scale aerial images, the objects such as buildings are big and rather homogeneous. We expect those would not so much affect the overall performance. Beside that, the hand labeling for the ground truth is done after downscaling and the comparison is performed on pixel level on these images. We use two folds cross-validation for learning and testing: six images are using for learning, six other images are used for testing and vice-versa.

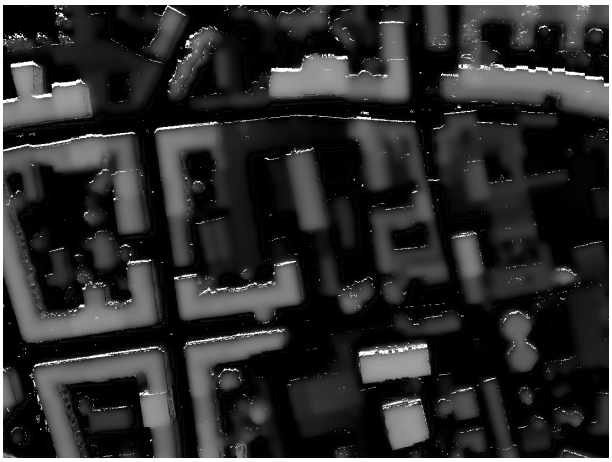
3.4.2 Feature Types

Our goal in the experiments is to show the performance of the hierarchy model by incorporating multiple classification potentials of different feature types and spatial (contextual) relationships. Thus, we simply use a standard feature computation for each data source to get feature types for the evaluation.

(a)



(b)



(c)

Figure 3.3: A typical scene taken from the data set: (a) color image, (b) hand-labeled buildings mask (c) the corresponding relative 3D height information.

-	Image data source	Feature types	Extracted feature vector
1	RGB	Color feature	R, G, B, mean, std, ratios between R, G, B
2	Gray scale	Texture feature	Gabor filter with 2 freq. and 4 orients
3	Height field	Height feature	Pixel value, mean and Std

Table 3.1: Image data sources and the used feature types

There are two sources of image data available, those are the RGB images and the corresponding height filed data. There are three different feature types extracted from these sources, which are color feature, texture feature and height feature.

For the RGB image data as visual cue, two types of features are extracted: Color and texture. The RGB feature type comprised of: the single pixel values of all input planes (R, G and B); the ratios between image planes (this is to remove the influence of light and shadow on a ridge due to the sun angle); the mean and standard deviation computed in a squared neighborhood of size 5x5 pixels surrounding the pixel. This results in a 9-dimensional feature vector for each pixel.

For the texture feature, we use Gabor filter for its well performing in edge detection and representation of texture features. First, the RGB image is converted to a gray scale image. Then the Gabor filter with two filter frequencies and 4 filter orientations is applied to the gray image. This gives a 8-dimensional feature vector for each pixel.

For the height field data, for each pixel we extract 3-dimensional feature vector, which represent for the height value of the point, the mean and the standard deviation in a squared block of size 5x5 centered at the pixel.

The use of neighboring pixels to compute features for each feature type explicitly includes neighboring context of the data into the local classification potential. A summary of image data sources, feature types and feature vector components for each feature type are given in Table 3.1.

3.4.3 Performance Evaluation

The upper part of Figure 3.4 shows a typical image patch of a complex urban scene. A zoomed version of a sub-patch of the RGB image is shown in the middle lower (b). The detection of buildings regions given by a traditional classifier is shown on the left hand side (a) and the performance of our model is shown on the right hand side (c).

The detection is depicted as raw outputs of each model, without a post processing step. As one can see most of building pixels are detected and building's boundaries are delineated.

Figure 3.4: Example of a large aerial image patch (upper part) with a small zoomed patch (b); Detection result of traditional method (a) and of our HpCRF model (c) (a post-processing step is needed to remove some noisy).

There are some areas in which the detector is confused and gives weak detection with some missing pixels, e.g. some areas in upper part of (a). These are hard points, which are usually low-building areas and/or teh building regions with similar appearance as street layer. Some of these missed detections may be recovered using neighboring information as context in a conventional MRF or CRF model. However, with a regular model of context for neighbor smoothing, some of these may get worse. For example, some missed detection holes on rooftop could get recovered. But some area of a small, low building in the upper-middle, where the "black points" (negative class) is more densely distributed, the "white points" (positive class) will be eroded. Which may cause severe distortion of building's delineation. Our model overcomes this. Some remaining noisy on the street can be cleaned by a post processing step.

For a **quantitative evaluation**, the detection rates in term of Recall, Precision and F-measure of different feature types and different models are shown in Table 3.2 and Ta-

Num.	Feature types	Recall	Precision	F-measure
1	Color only	74.6	84.1	79.1
2	Color & Pots	**83.8**	**91.4**	**87.4**
3	Height only	79.0	82.1	80.5
4	Height & Pots	**84.7**	**87.6**	**86.1**
5	Texture only	69.3	74.9	72.0
6	Texture & Pots	**84.9**	**86.4**	**85.6**
7	Mixture of features	**84.1**	**87.3**	**85.7**

Table 3.2: Classification performance of SVM on different feature types and their incorporation with other classification potentials. *Pots* denote the classification potentials from the classifier on other feature types

ble 3.3.

In Table 3.2, lines 1, 3 and 5 show the performance of a discriminative classifier (SVM) on each of the three individual feature types. The values are averaged over all test images. Lines 2, 4 and 6 show the performance of the same classifier on each of these feature types where the feature vector is expanded with the classification potentials from other feature types. I.e. the feature vector of the color feature is augmented with prediction confidences of classifier on texture and height features, and so on. The last row shows the performance of the classifier on mixture of all kind of features, i.e. feature vectors of the three types are concatenated.

The results show significant improvements of the detection rates on features expanded with classification potentials from other feature types. This is obvious since more useful information is provided to the classification. The notable point is: all these features expanded with potentials from others outperform, or at least as good as, the performance on mixture of features. Note that *the same source of input information has been used*. This supports our argument that, decompose the input feature space into different feature types, train separate classifiers for each type then incorporate the classification potentials at higher level would leverage the classification results.

Table 3.3 presents the performance of several models, include the discriminative classifier (SVM) on mixture of features, single CRF on the mixture of features, and our HpCRF. The values are averaged over the test images. All models are built using the SVM as the base classifier. As one can see, our system gives better results over the state-of-the-art traditional methods (SVM and CRF). If ones just have a look at the improvement in term of such quantitative values, it seems not that impressive! However, with a powerful classifier (like SVM) as the base learner and

Model types	Recall rate	Precision rate	F-measure
SVM	84.1	87.3	85.7
Single CRF	84.6	90.9	87.6
HpCRF	**85.2**	**96.3**	**90.4**

Table 3.3: Classification results of different models on the same input data.

good image features, getting some improvement of accuracy is not an easy task! The improvement here, even just few percents of accuracy, is really significant as it is done at the difficult points, where most of the conventional classification methods failed.

We have conducted extensive set of experiments on various combinations of feature types and classification potentials at different levels of interactions. In the following, we will show some visual illustration of the performance of the discriminative classifier with different image feature types and the performance of different classifiers on all features. For a clearance, Figure 3.5 shows a zoomed version of a typical aerial image patch from upper part of Figure 3.4 and the corresponding labeled ground truth.

First, we will show the performance of the traditional classifier on different feature types (color, texture and height) and on the mixture of these features. As it can be seen in Figure 3.6, when all features are used, the classifier gives the best result over individual feature types. This is clear as more information (as feature cue) is provided to the classification.

Second, given the ground truth and the performance of the traditional classifier on all features in Figure 3.7, we hight light the performance of HpCRF over traditional classifiers.

We will show a trivial case: applying some morphological operators on the classification results obtained by the traditional classifier (SVM) trained on all features. These are morphological operators for binary image given by Matlab functions *imdilate, imerode* and *imopen*. We get the results as shown on Figure 3.8. As one can see, these operators clean out most of false positives (e.g. noisy detection on the street), but also make serious mistake in removing building pixels (i.e. the *erosion* or the *morphological opening* operators), or adding more missed detections (using the *dilation* operator).

The overall performance comparison is depicted in Figure 3.9 and Figure 3.10. Figures 3.9(a) and 3.9(b) are the detection results given by the local classifier on the color feature and the height data feature types, respectively. One can see that for the color feature type, the classifier makes wrong decision in the building's regions which have similar color

and low contrast to the ground. For example, the region marked as **[2]** in 3.9(b) is almost missed in 3.9(a). For the classification using the height data feature, some height trees are detected as buildings, but some low building regions are missed. For example, the region marked as **[1]** in 3.9(a) gets lost as in 3.9(b).

By concatenating all feature types together, the classifier makes better detection results: as in Figure 3.9(c), both regions **[1]** and **[2]** get detected. The region **[1]** is clear, but the region **[2]** is rather weakly detected. Besides, there are still a lots of false positives, such as cars, trucks or other noisy on the ground.

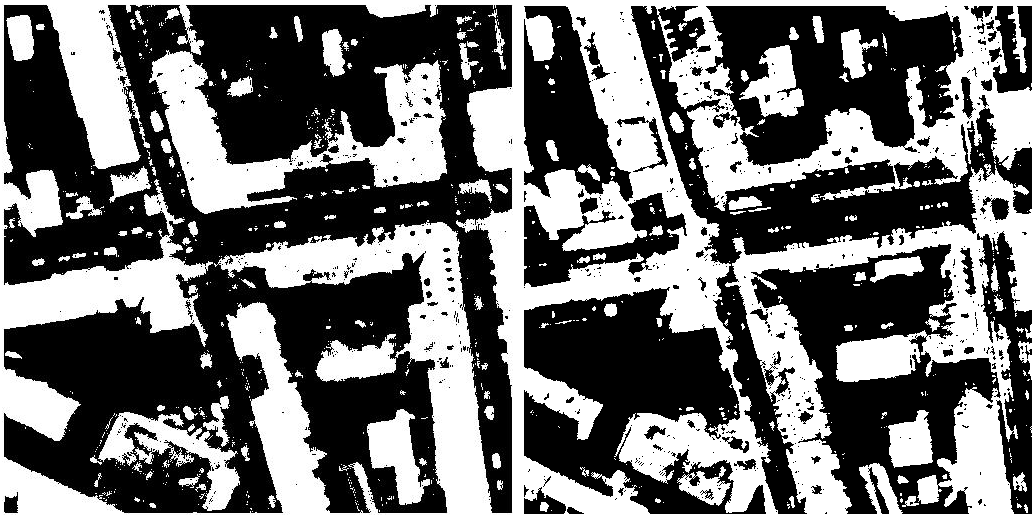
By incorporating the contextual information in a random field model, one could get improvement: various noisy objects have been removed, some "black holes" caused by missed detection on building's rooftops can be "healed". The result is shown on the Figure 3.10(b). One can see that the region marked **[3]** and **[4]** get recovered, but some small building regions such as **[1]** and **[2]** almost get lost. This is an unexpected result.

By using our HpCRF model, we obtain the detection result as shown on Figure 3.10(c). One can see that both regions marked as **[1]** and **[2]** are now get detected. Some remaining noisy objects on the ground can be easily cleaned at a post processing step. The detection result of the HpCRF is the most accurate approximation of the ground truth, as shown on Figure 3.10(d).

3.5 Conclusions

We have proposed a new hierarchical pseudo conditional random field model for learning complex object class, i.e. buildings from aerial image. The model decomposes the input feature space into different feature types for training different probabilistic classifiers. The features and the classification potentials are then integrated into a unified hierarchical probabilistic model. The hierarchy structure of the model allows to exploit the discriminative power of each feature type and to leverage the performance by using spatial context and integration at higher levels. The proposed framework provides a simple yet efficient way to model complex object class. Learning and inference are efficient, general and straight forward. It can be easily implemented and applied to a number of visual learning tasks. The experiments show the successful application of our model on a real world application of detection and segmentation of buildings at pixel level on large scale aerial images. The results show the improvement of the proposed model over traditional stat-of-the-art approaches.

Figure 3.5: A typical RGB image patch (upper) and the corresponding ground truth (lower).



(a) Classification using color feature (b) Classification using texture feature

(c) Classification using height feature (d) Classification using all features

Figure 3.6: Classification of building class using the traditional classifier (SVM) on different feature types: (a) when only color cue is used, the building pixels with similar color to the street layer get lost, some objects on the street are wrongly classified as building; (b) similarly when only texture information is used; (c) when only height data is use, low buildings get lost and height trees are classified as building; and (d) when all features are used, the classifier gives the best result.

(a) Labeled ground truth

(b) Classification result of SVM on all features

Figure 3.7: The given ground truth (buildings masks) and the performance of the discriminative classifier (SVM) on mixture of features.

(a) Classification result using all features

(b) Result of (a) with dilation operator

(c) Result of (a) using erosion operator

(d) Result of (a) with morphological opening

Figure 3.8: A simple comparison: applying the morphological operators on classification results (no context information is used in the classifier.

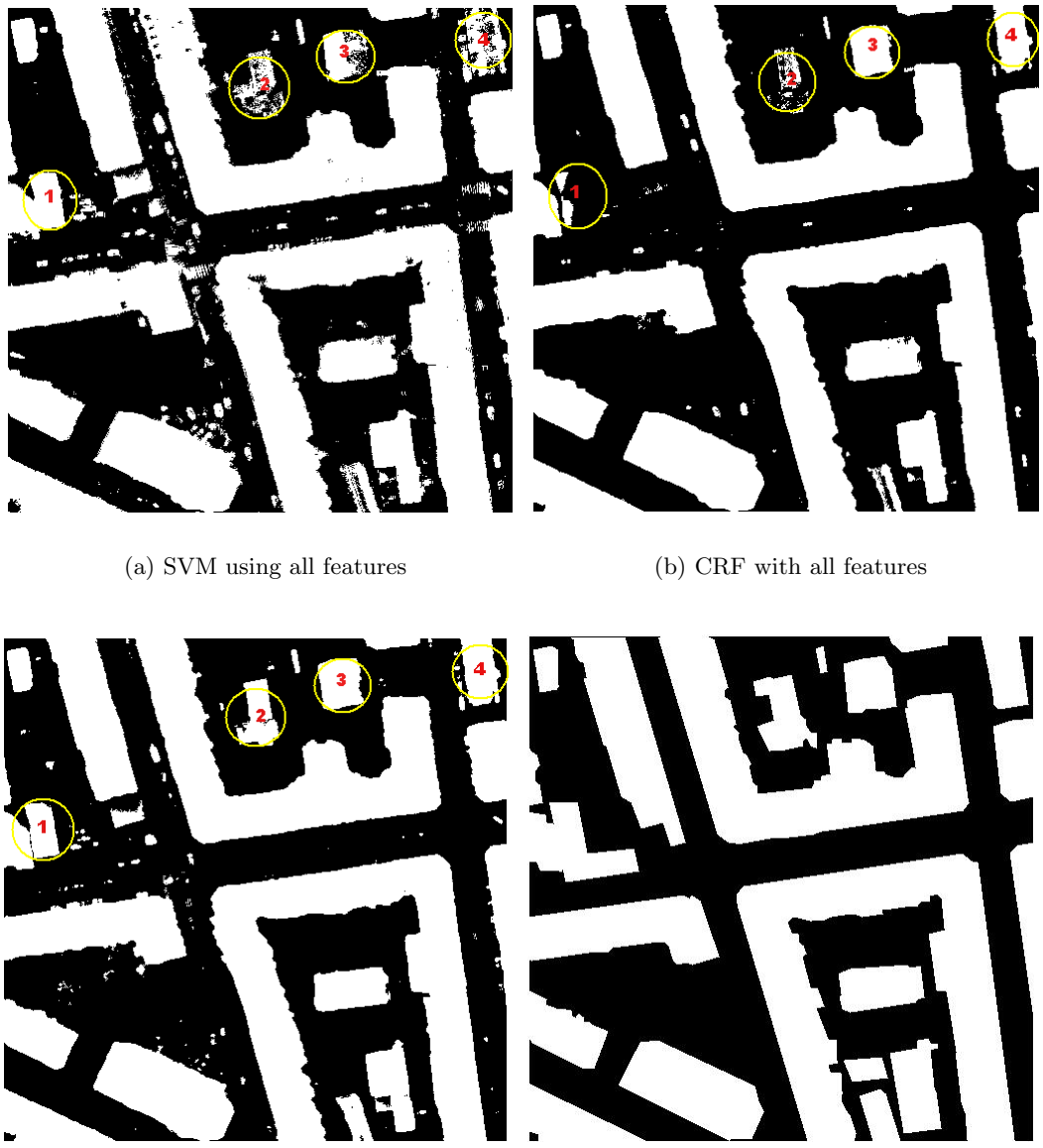
(a) Classification using color feature

(b) Classification using height feature

(c) Classification using all features

(d) Classification using CRF

Figure 3.9: Classification of building class using SVM classifier on different feature types ((a) and (b)), on all feature types (c), and classification result of the CRF (d).



(a) SVM using all features

(b) CRF with all features

(c) HpCRF with all features

(d) Ground truth

Figure 3.10: Classification results of different methods using all feature types. The traditional state-of-the-art classifiers: (a) SVM with an accuracy of 85.7%, (b) Single CRF with accuracy of 87.6%. Our HpCRF model (c) with the best result of 90.4% (see also the text for more details).

Chapter 4

Conclusions and Future Work

Contents

4.1 Summary of main contributions

In this thesis, we have addressed the problem of object detection from aerial images, including the regular object class (car) and the complex object class (building). We have developed several learning frameworks for the detection of object of the two categories. Toward this, the thesis have made the following key contributions:

- We have developed a learning framework based on the online boosting for feature selection method for learning the regular object class, i.e. car. The system allows the use of efficient representative features for data representation. We have demonstrated that fully supervised interactive training allows effective learning and improvement of the classifier. Robust performance has been obtained with fast and efficient processing for detection of cars on large scale aerial images. The system requires no prior knowledge of the object model as well as no labeled training data in advance. All of these crucial knowledge of the model are acquired during the online learning process. The framework has been developed focusing on the detection of cars, but for other objects in the category the approach is applicable.

- We have proposed to extend the online boosting based learning framework with an active learning procedure. The new framework allows to exploit the availability of classifier during learning to automatically generate training samples "on-the-fly": The training classifier is applied on coming image to detect the object and to bootstrap the false positives as negatives samples to feedback to the training. This addresses the issue of reducing labeling effort meanwhile obtain better performance by gradually improving the classifier. We have proved experimentally that active learning based on an online boosting approach trained in this manner can achieve results comparable or even outperform a framework trained in conventional manner using much more labeling effort.

- We have proposed a novel probabilistic model: HpCRF - A Hierarchical Pseudo-Conditional Random Field - based on the principles of conditional random field and graphical model. The HpCRF is constructed as a hierarchy of multiple image characteristics to capture different aspects of image data and the spatial dependencies between observed data and labels. Each layer is composed of multiple CRFs. At the first layer, each CRF is responsible for a certain image property and its spatial context. This aims at exploiting discriminative power of each feature type. At the second layer, each CRF attempts to exploit the inter dependencies among different feature types and predicted confidence of related labels. The main advantage of the proposed model is its ability to incorporate mutual dependencies among different aspects of image data as well as their spatial context information in a sound principled manner.
  This is the first hierarchical random field model which works on the decomposition of image properties, exploits the inter-feature and spatial dependencies between image data and the prediction confidences, and then combines them at later stage to leverage the performance of the traditional approaches in classifying the object class.

- We have developed a learning algorithm for the HpCRF model. The algorithm can be considered as a meta learning scheme, in which the learning is performed sequentially for each layer. Any discriminative classifier which can give probabilistic output can be employed for learning the local classification potential for each individual CRF. Learning of individual CRF is performed by a pseudo strategy. Inference is done by a non-iterative Iterated Conditional Modes (ICM) method to get classification confidence for intermediate layers, and to get object class for the final output. Learning and inference are simple yet effective and general that can be easily implemented

and applied for several learning tasks.

- We have shown the applicability and the efficiency of using the online boosting based framework for the car detection problem. We have demonstrated the applicability and the benefits of the HpCRF model in the application for the problem of detection and segmentation of buildings from aerial images.

4.2 Discussion

In this section, we will discuss aspects such as the possibilities of the proposed systems for other applications, the limitations as well as the open issues of the proposed frameworks.

On the online boosting based framework

- For the online boosting for car detection framework, the following issues are concerned: One of the main advantages of the aerial digital camera, i.e. the *UltraCamD*, is its ability to deliver multi-spectral images with high overlapping areas. The high redundancy of overlapping images could be exploited in different manners to improve productivity of the system. In our framework for the car detection, we have not yet exploited the redundancy explicitly. One way could be, as in a later work, in the similar spirit of the problem for regular object detection (car), Kluckner et. al [68] has extended the work by using the 3D high data derived from high resolution aerial images as a "teacher" for training a car detection system. The work obtained similar result with reduced hand labeling efforts.

- However, we have made a further step to utilize the framework for learning, detection and tracking of other objects, including deformable object, e.g. hand postures. Part of our publications number (3) and (5) (see Appendix B) demonstrated the fact. These support our argument that the developed framework can be used for learning and detection of other objects, from aerial as well as terrestrial image, or from video sequences.

- For the learning in the framework, there is no need to build the labeled training data set in advance. The process of supervised learning with interactive training shall generate samples during the learning process. When the learning finished, beside the "main product", i.e. the obtained classifier, one also gets a set of labeled samples as a "by product". This brings us to a new application: to extend the framework

for learning the object model meanwhile generate informative object samples (either positive or negative), which can be saved as object database for other training process (benchmark set) or a transfer learning. This has been demonstrated partly in our publication number (2) (see Appendix B).

On the HpCRF model

For the proposed hierarchical pseudo conditional random field, the following issues need further discussion:

- Tu [182] employed the idea of expanding a sample's features with related predictions (classification confidences). However, there was no decomposition of features. In his approach, a single layer CRF model has been constructed with an iterative training. While we use feature decomposition in a hierarchy model structure using the pseudo-training strategy. The most similar work in decomposition of feature cues for learning a random field model is the work by Ma and Grimson [101], where features are modeled by a contour process and texture process for learning a coupled conditional random field (see also Section 3.2.5).

- We set our goal to develop a novel hierarchy model for the problem of object categorical classification of the building object class, and not aim to the ultimate goal of the whole aerial scene interpretation, e.g. Porway et. al [144]*.

- Up to our knowledge, this is the first hierarchical probabilistic random field model in computer vision which models the mutual dependencies among different properties of image data, in which the classification potentials, the spatial dependencies between labels (in term of predicted confidence) and observed data are learned in a consistent framework.

- In the current version of the proposed model, we have used a regular grid with direct interactions for spatial context. Meanwhile, long range interactions (higher order of clique or wider window on the grid) could be also used to provide more contextual information.

- There should be more investigation on modeling context for each feature type, which represents different aspects of data as each feature cue may have different context

*Note the difference in this hierarchical contextual model and our model, i.e. they used the hierarchy model at object and scene levels while we used the hierarchy structure at low level of features.

to exploit (pixels which have similar color may belong to the same class, however, pixels with the same hight values may belong to either building or high tree).

- For the learning and inference algorithm: The procedure for estimation is a bit ad-hoc, since different layers of the models are not really jointly learned. The fact is that, due to the hierarchy structure of the model, the upar layer depends on the outputs of the lower layer, learning has to perform sequentially. Besides, it was our attempt to design the model for a particular problem of buildings detection and segmentation.

- Multiple kernels would be helpful in weighting the contributions of each source of information, i.e. each type of the features. Thus, multiple kernel learning should be also investigated.

- The proposed model has been validated on large scale aerial images data. However experiments have not yet conducted on various generic object detection/recognition data sets.

- At the time of this work, there are continuing number of research works in investigating methods for exploiting context in the random field modeling approach [95, 126, 142, 158].

  The works of [95] and [142] are in favor of using the hierarchy of regions produced by a generic segmentation method in order to model context. Plath et. al [142] extended the figure-ground segmentation work of Reynolds [149] by considering image classification as global feature to couple with local image features in a CRF. Lim et. al [95] used context by region ancestry: hierarchical segmentation tree is constructed, where each leaf is described by features of its ancestral set (the regions on the path linking the leaf to the root). Nowozin and Lampert [126] presented a global potential function by enforcing connectedness of the output labeling. Finally, Divvala et. al [158] presents an empirical study of the role of context in object detection, the sources of context and the ways to use it. The work reaffirmed that contextual reasoning is a critical part of the object recognition problem.

4.3 Future work

Several limitations of the proposed framework have been discussed in previous section. The future work might address these limitations as well as to explore further model extensions.

- For the online boosting based learning for the car detection framework, the following issues could be further investigated:

  Diversification of the features or parameters for weak classifiers could increase the complexity of the system making it possible to deal with hard samples.

  The use of information from aerial triangulation and possibility also dense matching to detect cars in multiple, overlapping images that differ in their viewing angle including automatic combination of the results. This yields higher performance for the system.

  The framework could also be improved by exploiting contextual information from neighboring patches in a CRF fashion.

- Possible improvements for the hierarchical HpCRF model:

  The used features in the current model are quite simple and generic. More powerful representative features should be investigated.

  In this work, we used simplifying assumptions for parameter learning and inference in the hierarchical framework. In the future, it is essential to explore the learning process in several aspects:

  For the local (base) classifier of individual CRF, beside SVM, some other powerful discriminative classifiers, such as randomized forest or boosting could be employed and compared.

  The spatial dependencies just have been modeled on regular grid with direct connections. More complex relationship (at region or global scale, e.g. [95, 126]) should be studied for modeling the contextual potential.

  Multiple kernel weighting must be addressed to investigate the influence of different feature type to the entire model performance.

  The model should be validated on more aerial images and more challenging data sets.

  Semantic classification in aerial imagery as well as terrestrial images is an interesting topic to explore the learning and the performance of the model.

Appendix A

Object Recognition Overview

Contents

The review of close related issues are presented in each chapter for each particular topic. In this section, we briefly give an overview of general systems and popular approaches for object recognition over time of development of the field.

Object detection/recognition has been being an intensive research topic in computer vision for several decades. Over the years, this research area has been rapidly developed with advances and efforts of researchers in the field. This involves various aspects, such as the developments of complex mathematical tools and modeling techniques, deeper understanding in the visual perception research, increasing computational power for implementation of powerful learning algorithms, or more challenging data requirement.

There have been a number of publications addressed issues regarding evolution of research in object categorical recognition, i.e. Pinz [141], Grabner (Chapter 1.) [41] and very recent work of Dickinson [32]. In this section, we sketch out major components of an object recognition system and summary several landmarks for object recognition from the early years to the current state of the art. We do not attempt to render a complete review of history and current efforts on the subject. The goal here is just to provide a general picture of the object recognition research field, which aims to guide the readers to put the methods developed in this thesis into the context.

A.1 Fundamental Issues of Object Recognition

We are dealing with generic object recognition problem (also called object categorization), which is a process of assigning a specific object to a certain category. Examples of categories in generic object recognition are people, horses, buildings, cars, bikes, etc. This is in contrast to specific object recognition, which deals with the recognition of a specific, individual object, like Mr. president, my bike [141].

Every day we recognize a various regular as well as novel objects. Human beings do this with little effort, despite the fact that these objects may vary somewhat in form, color, texture, etc. Objects are recognized from many different viewing points (from the front, rear, or side), under many illumination changes, at many different places, and in different scales. Objects can even be recognized when they are partially cluttered or obstructed from view. However, the "progress in understanding the brain mechanisms underlying vision requires the construction of computational models that not only emulate the brain's anatomy and physiology, but ultimately match its performance on visual tasks.", and therefore "Visual object recognition is an extremely difficult computational problem" [140].

There are a number of fundamental issues of an object recognition system. Major issues that have been addressed intensively are: *Object representation, Learning and Recognition*. Further issues should also be taken into account, such as object localization, database, and performance evaluation [141]. In the following we will briefly discuss the major issues.

Object representation addresses the questions: how visual object category are represented? what features should be extracted and how are they described? There are many ways an object can be represented. Geometric primitives such as edges, lines, polyhedra, or high order statistical object properties can be used as features. The representation has to cover properties of object in the image, such as: color, texture, contour, regions, shape, topology or even representative function of an object. Features can be extracted at pixel level, local image patch, or entire image. The descriptors should be invariant against potential distortion of radio metric or geometry, intra-category variations and robust to discriminate the object class.

Learning: Machine learning and computer vision are the two strongly related fields. Over decades, machine learning techniques have been using extensively for solving many computer vision problems. At early time, recognition was done by matching extracted

features with the object model [80, 152]. Modern approaches tend to use machine learning techniques to learn the object model from a set of training examples [36, 53]. After extracting representative features from training examples, learning process is performed. The recognition problem can be formulated as classification problem and the system learns the object classes (i.e. categories). Learning process can be performed in several fashions: supervised or unsupervised, off-line (batch) or on-line learning. Different learning models can be used, i.e. discriminative, generative, or combined of both. Typical available learning methods are neural network [50], Bayesian network [164], SVM [130], boosting [129, 187], and random forest [90, 167].

Further issue should be addressed is the use of contextual information to improve the performance of traditional learning model.

In our work, for the first object category, we learned a discriminative model, where Boosting learning technique has been used in an online version; for the second object category, we built a hierarchical contextual model with a meta learning scheme, where any probabilistic discriminant classifier can be employed.

Recognition (classification): after learning, the resulting model is called classifier (or detector). The recognition process is carried out by applying the classifier on novel test data. Depending on the representation, classification output may be pixel label (segmentation), label of each image site (bounding box), or image class. The performance of the system is evaluated at this stage. Depending on the learning method, the trained model can perform continuous learning to adapt to unseen samples (online), or fixed (offline).
In the followings, we outline some landmarks of object recognition research, which pave the ways to current state-of-the-art research of the field.

A.2 Object Recognition System Overview

Figure A.1 gives an overview of the Evolution of object recognition over the past four decades [32].

Early time of computer vision

The early years of computational object recognition dated back to 1960s. We direct the readers to the work of Grabner [41] (Chapter 1) for a historical view of publications in related journals and conferences.

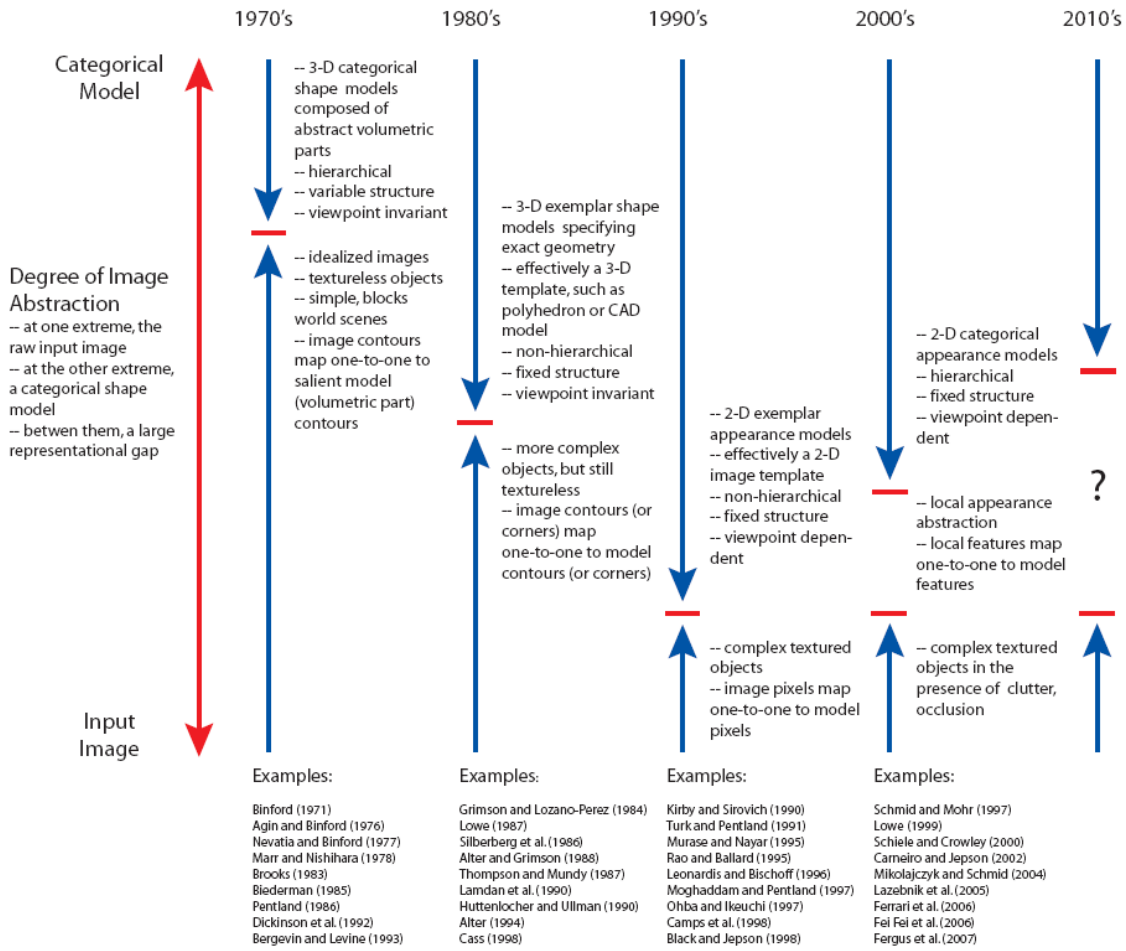


Figure A.1: The evolution of object categorization.

At the early stage, i.e. 1970s, the object recognition system was mainly model-based approach. Object's shape and appearance were used in an explicit model of to represent the object of interest. The approaches made simple assumptions about the real visual scenes for Model Identification, e.g. "Let us say that we are given a picture of a parallelepiped and it has been reduced to a line drawing" [152].

Many of the approaches used high level object's primitive entities, such as edges, boxes, cylinders, or boundaries Geometric constraint and topological/spatial relationships were applied for recognizing objects in scenes. These methods have advantages in insensitiveness to illumination changes and capability of recovering 2D and 3D poses.

The "Blocks World" of [152] assumed that objects of interest such as blocks, prisms

were made of combinations of polyhedra and appeared in a uniform background. Edges and lines were detected as features. Recognition was done with matching the polygon structures to the models by topological constraints.

Model-based object recognition systems can be roughly divided into two categories: object-centered representation and view-centered representation. One of the early remarkable work is of D. Marr [104], where the idea has influenced a decade of research, advocating 3D reconstruction and modeling of objects in the scene.

The ACRONYM [16] system based on 3D models to interpret 2D images. Three-dimensional geometric objects were modeled as generalized cones and their spatial relationships; edges were combined into features such as ribbons and ellipses. Interpretation proceeded by combining local matches of shapes into more global matches, requiring consistency among matches.

In the 1980s, the recognition systems were mainly 3D models represented by 3D templates of objects. These exact shape models were inspired by CAD models. Up to this time, due to the search complexity, the objects were still modeled as texture-free. Biederman [14] proposed the *recognition by components* theory, where the volumetric primitives, so-called *geons* can be used to recognize objects in a generic way.

In [46], the objects were modeled as polyhedra and sets of planar faces. Objects are identified and located in a scene by matching positions and surfaces to those in 3D models. The method proceeded by examining all hypotheses about correspondences between sensed data and object surfaces.

In SCERPO system proposed by Lowe [97], pairs of straight lines were combined into perceptual structures. A process of perceptual organization was used to form groupings in the image, then these primitive structures were combined into larger, more complex structures such as trapezoid shapes.

Huttenlocher and Ullman [61] presented two stages model-based method for recognizing solid objects with unknown 3D position and orientation from a single 2D image. In the first stage, possible alignments were computed to generate transformations from the model to the image. Local features derived from corners and inflection points were used for the computation of possible alignments. In the second stage, each of these hypothesized matches was verified by comparing the complete edge contours of the aligned objects with the observed image edges.

In this period, the models have been brought closer to the imaged objects. However, the objects were still modeled as textureless, and objects with complex surface markings

were unable to be recognized.

Global appearance methods

In the late of 1980s and early 1990s, there has been a transition from 3D model based representation to 2D multi-view based representation: using a set of 2D views to describe and recognize 3D objects. The early methods objects are represented by storing their global appearance information. Viewpoint and lighting invariance were achieved by capturing images from many viewpoints and under various illumination. Recognition in a new image was done by finding the most similar image in the stored database.

Ullman [184] represented a 3D object with the linear combination of 2D images of the object. The idea was that visual object recognition requires the matching of an image with a set of stored models. Vetter and Poggio [112] proposed a method to generate virtual new views given one view of an object by exploiting prior knowledge. An example-based approach was used as an alternative to 3D model-based approach. Linear combination of views were used to synthesize new views.

The global approaches model the information of the whole image. This results in high dimensionality of representation of data. Therefore, subspace methods are widely used. The main idea is to project the original input images representation onto a suitable lower dimensional subspace. The projection has to retain as much as possible information that represents the data best for a specific task. Principal Component Analysis (PCA) [65] is the most prominent subspace method. Other global linear subspace methods include independent component analysis (ICA) [62], linear discriminant analysis (LDA)[105].

Turk and Pentland [183] used an approach of eigenspace to the detection and identification of human faces. Murase and Nayar [116] extended the eigenspace method to recognize 3D objects.

Local appearance methods

The global appearance method has advantages in its simplicity and computational efficiency. However, the methods are sensitive to background clutter and occlusion, and usually require large amount of training data. From late of 1990s onward, research in object recognition has made movement to utilize local appearance information to recognize objects. The general idea of using local methods is to represent objects by the

appearance of a set of single points, local regions or patches extracted from the images. Local approaches search for locally salient regions, which are characterized by a proper descriptor.

Recognition typically proceeds by matching the local regions in new images to the local regions of model objects in the database. Geometrical modeling can be additionally used to obtain more discriminative power.

Schmid and Mohr [163] proposed a seminal work using a collection of automatically detected local regions to represent objects. Interest points were extracted using a Harris corner detector [52]. Each local region around an interest point was described by a vector of rotationally invariant gray-scale measures. Object image retrieval was carried out with a voting algorithm and semi local constraints.

Local descriptors have to be extracted at the salient locations. Possible descriptors are calculated from a local support region and can for instance be moments of various order or intensity distributions. The Scale Invariant Feature Transform (SIFT) plays a special role [98]. This method closely couples a difference of Gaussian (DoG) keypoint detector with SIFT as a local description method. A good comparison of local descriptors can be found in [111].

The idea of using local regions, appearance based methods has strongly influenced state-of-the-art approaches in object recognition research [129]. There are two main types of modeling in these approaches: methods with geometric constraints and methods with geometric-free. Constellation models [17, 37, 38, 191], sparse, part-based representation models [2] are of the former type; Bag-of-keypoints [27], Bag-of-words [170], random subwindows [102] are of the later type.

We direct reader to a good reference for appearance based methods for object recognition in a recent survey of Roth and Winter [154].

In many recent works, advances topics such as: online, active learning, semi-supervised learning [129, 195], exploiting contextual information [19, 34, 56] to improve the performance of any existed detector, etc. have drawn a lots of research interests.

The work in this thesis is on the problem of object detection/categorization in aerial images. Many components in our proposed systems for detection of objects of the two categories are built upon various precedent successes. The employing of robust machine learning methods and the proposal of novel model architecture has been demonstrated by our better object categorical recognition systems.

Appendix B

Publications

During my research work at the Institute for Computer Graphics and Vision, Graz University of Technology, the following papers have been were finished. This thesis is mainly based on some of these publications. For the sake of completeness, the papers are listed in an inverse chronological order.

(1) Thuy T. Nguyen and Horst Bischof, "HpCRF: A Hierarchical Pseudo-Conditional Random Field Model for Buildings Detection from Aerial Image", Submitted to *The IEEE International Conference on Computer Vision (ICCV)*, Japan, 2009.

(2) B. D. Nguyen, Thuy T. Nguyen, "Automatic Database Creation and Object Model Learning", *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 5465/2009, p. 27-39. May 2009.

(3) Thuy T. Nguyen, B. D. Nguyen and Horst Bischof, "An active boosting-based framework for real-time hand detection", in *Proc. of The 8th IEEE International Conference on "Automatic Face and Gesture Recognition (FG08)"*, Amsterdam, Sep. 2008.

(4) Thuy T. Nguyen, B. D. Nguyen and Horst Bischof, "Efficient boosting-based active learning for specific object detection problems", in *Proc. of The 5th International Conference on "Computer Vision, Image and Signal Processing (CVISP 2008)"*, Praha, Jul. 2008.

(5) B. D. Nguyen, Thuy T. Nguyen and Horst Bischof, "On-Line Boosting Learning

for Hand Tracking and Recognition", in *Proc. of The 2008 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'08)*, Las Vegas, USA, Jul. 2008.

(6) Helmut Grabner, Thuy Thi Nguyen, Barbara Gruber and Horst Bischof, "Boosting-based car detection from aerial images", ISPRS, *Intl. Journal of Photogrammetry and Remote Sensing*, Vol. 63/3, p. 382-396. DOI information: 10.1016/j.isprsjprs.2007.10.005.

(7) Thuy T. Nguyen, Helmut Grabner, Barbara Gruber and Horst Bischof, "On-line boosting for car detection from aerial images", in *Proc. of The 5th IEEE International Conference on "Research, Innovation and Vision for the Future (RIVF'07)"*, Hanoi, Mar. 2007 (Best paper Award).

Bibliography

[1] Abramson, Y. and Freund, Y. (2005). SEmi-automatic VIsual LEarning (SEVILLE): Tutorial on active learning for visual object recognition. Technical report, UCSD.

[2] Agarwal, S., Awan, A., and Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490.

[3] Al-Bakri, J. T., Taylor, J. C., and Brewer, T. R. (2001). Monitoring land use change in the badia transition zone in jordan using aerial photography and satellite imagery. *The Geographical Journal*, 167(3):248–262.

[4] Alba-Flores, R. (2005). Evaluation of the use of high-resolution satellite imagery in transportation applications. Technical report, Intelligent transportation system institute, University of Minnesota.

[5] Baltsavias, E. (2002). Object extraction and revision by image analysis using existing geospatial data and knowledge: state-of-the-art and steps towards operational systems. *ISPRS Journal of Photogrammetry & Remote Sensing*, 58:3–4.

[6] Baltsavias, E. (2004). Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. *International Journal of Photogrammetry and Remote Sensing*, 58:129–142.

[7] Barczack, A. L. C., Johnson, M. J., and Messom, C. H. (2005). Real-time computation of haar-like features at generic angles for detection algorithms. *Research Letters in the Information and Mathematical Sciences*, 9:98 – 111.

[8] Beleznai, C., Fruhstuck, B., Bischof, H., and Kropatsch, W. (2004). Detecting humans in groups using a fast mean shift prozedure. In *Proceedings Workshop of the Austrian Association for Pattern Recognition*, pages 71–78, Hagenberg, Austria. Austrian Computer Society (OCG).

[9] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522.

[10] Berger, A. (1997). The improved iterative scaling algorithm: A gentle introduction.

[11] Bernstein, E. and Amit, Y. (2005). Part-based statistical models for object classification and detection. In *Proceedings of Computer Vision and Pattern Recognition*, volume 2, pages 734–740, San Diego, CA, USA. IEEE Computer Society.

[12] Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–236.

[13] Besag, J. (1986). On the statistical analysis of dirty pictures. *Royal Statistical Society*, B-48(3):259–302.

[14] Biederman, I. (1985). Human image understanding: Recent research and a theory. *Computer Vision, Graphics and Image Processing (CVGIP)*, 32:29–73.

[15] Bileschi, S. M., Leung, B., and Rifkin, R. M. (2004). Towards component-based car detection. In *ECCV Workshop on Statistical Learning and Computer Vision*, pages 75–98, Prague, Czech Republic. Springer.

[16] Brooks, R. (1981). Symbolic reasoning among 3d models and 2d images. *Artificial Intelligence*, 17:285–348.

[17] Burl, M. C., Weber, M., and Perona, P. (1998). A probabilistic approach to object recognition using local photometry and global geometry. In *Proceedings of The 5th European Conference on Computer Vision*, pages 628–641.

[18] Butler, D. (2006). Virtual globes: the web-wide world. *Nature*, 439:776–778.

[19] Carbonetto, P., de Freitas, N., Freitas, O. D., and Barnard, K. (2004). A statistical model for general contextual object recognition. In *Proceedings of the European Conference on Computer Vision*, pages 350–362.

[20] Champion, N. (2007). 2D building change detection from high resolution aerial images and correlation digital surface models. In *Photogrammetric Image Analysis*, pages 197–202.

[21] Champion, N., Matikainen, L., Liang, X., Hyyppä, J., and Rottensteiner, F. (2008). A test of 2D building change detection methods: Comparison, evaluation and perspectives. *the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII:297–303.

[22] Champion, N., Stamon, G., and Pierrot-Deseilligny, M. (2009). *Lecture Notes in Geoinformation and Cartography*, chapter Automatic Revision of 2D Building Databases from High Resolution Satellite Imagery: A 3D Photogrammetric Approach, pages 43–66. Springer Berlin Heidelberg.

[23] Christensen, H. I. (2003). Cognitive (vision) systems. *ERCIM News*, pages 17–18.

[24] chung Chang, C. and Lin, C. J. (2001). LIBSVM: a library for support vector machines.

[25] Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

[26] Comaniciu, D. and Meer, P. (1999). Mean shift analysis and applications. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1197–1203, Kerkyra, Greece. IEEE Computer Society.

[27] Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Proceedings of The Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

[28] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, San Diego, CA, USA. IEEE Computer Society.

[29] Demiriz, A., Bennett, K., and Shawe-Taylor, J. (2002). Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254.

[30] Dexter, L. and Cluer, B. L. (1999). Cyclic erosional instability of Sandbars along the Colorado river, Grand Canyon, Arizona. *Annals of the Association of American Geographers*, 89(2):238–266.

[31] Dick, A. R., Torr, P. H. S., and Cipolla, R. (2004). Modelling and interpretation of architecture from several images. *Int. Journal Computer Vision*, 60(2):111–134.

[32] Dickinson, S., Leonardis, A., Schiele, B., and Tarr, M., editors (2009). *Object Categorization: Computer and Human Vision Perspectives*, chapter The Evolution of Object Categorization and the Challenge of Image Abstraction, page In press. Cambridge University Press.

[33] Dietterich, T. G., Ashenfelter, A., and Bulatov, Y. (2004). Training conditional random fields via gradient tree boosting. In *Proceedings of the 21th International Conference on Machine Learning (ICML*, pages 217–224. ACM.

[34] Divvala, S. K., Hoiem, D., Hays, J. H., Efros, A. A., and Hebert, M. (2009). An empirical study of context in object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[35] Drauschke, M. and Förstner, W. (2008). Selecting appropriate features for detecting buildings and building parts. In *Proceedings of The 21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Beijing, China.

[36] Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. Wiley-Interscience, 2 edition.

[37] Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *Proceedings of the CVPR Workshop on Generative Model Based Vision*.

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

[39] Fischer, A., Kolbe, T., Lang, F., Cremers, A., Forstner, W., Pluemer, L., and Steinhage, V. (1998). Extracting buildings from aerial images using hierarchical aggregation in 2d and 3d. *Computer Vision and Image Understanding*, 72(2):185–203.

[40] Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

[41] Grabner, H. (2008). *On-line Boosting and Vision*. PhD thesis, TU Graz.

[42] Grabner, H., Beleznai, C., and Bischof, H. (2005). Improving adaboost detection rate by wobble and mean shift. In *Proceedings of the Computer Vision Winter Workshop*, pages 23–32, Zell an der Pram, Austria. Austrian Computer Society (OCG).

[43] Grabner, H. and Bischof, H. (2006). On-line boosting and vision. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 260–267, New York, NY, USA. IEEE Computer Society.

[44] Greig, D. M., Porteous, B. T., and Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, 51(2):271–279.

[45] Grimson, W. E. L. (1991). *Object recognition by computer: The role of geometric constraints*. The MIT Press.

[46] Grimson, W. E. L. and Lozano-Perez, T. (1987). Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4):469–482.

[47] Gruber, M., Ponticelli, M., Bernögger, S., and Leberl, F. (2008). Ultracamx, the large format digital aerial camera system by Vexcel Imaging / Microsoft. *ISPRS Archives*, XXXVII. Part B1:665–670.

[48] Gruen, A. (1998). *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*. Birkhauser Boston.

[49] Gruen, A. (2008). Reality-based generation of virtual environments for digital earth. *International Journal of Digital Earth*, 1(1):88–106.

[50] H., R., S., B., and T., K. (1996). Neural network-based face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 203–208.

[51] Hammersley, J. M. and Clifford, P. (1971). Markov field on finite graphs and lattices. Unpublished.

[52] Harris, C. and Stephens, M. (1988). A combined corner and edge detection. In *Proceedings of the The Fourth Alvey Vision Conference*, pages 147–151.

[53] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning*. Springer.

[54] He, X., Zemel, R. S., and Carreira-Perpinan, M. A. (2004). Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages II–695–II–702 Vol.2.

[55] Heisele, B., Riskov, I., and Morgenstern, C. (2006). *Components for Object Detection and Identification*, volume 4170, chapter III, pages 225–237. Springer Berlin, Heidelberg, Germany.

[56] Heitz, G. and Koller, D. (2008). Learning spatial context: Using stuff to find things. In *Proceedings of the 10th European Conference on Computer Vision*, pages 30–43, Berlin, Heidelberg. Springer-Verlag.

[57] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

[58] Hinz, S. (2003). Detection and counting of cars in aerial images. In *Proceddings of the International Conference on Image Processing*, volume 3, pages 997–1000, Barcelona, Spain. IEEE.

[59] Hinz, S. and Stilla, U. (2006). Car detection in aerial thermal images by local and global evidence accumulation. *Pattern Recognition Letters*, 27(4):308–315.

[60] Hoiem, D., Rother, C., and Winn, J. (2007). 3D layout CR0f for multi-view object class recognition and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8.

[61] Huttenlocher, D. P. and Ullman, S. (1990). Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212.

[62] Hyvarinen, A., Karhunen, J., , and Oja, E. (2001). *Independent Component Analysis*. Wiley.

[63] Javed, O., Ali, S., and Shah, M. (2005). Online detection and classification of moving objects using progressively improving detectors. In *Proceedings Conference on Computer Vision and Pattern Recognition*, pages 695–700, San Diego, CA, USA. IEEE Computer Society.

[64] Jaynes, C., Riseman, E., and Hanson, A. (2003). Recognition and reconstruction of buildings from multiple aerial images. *Computer Vision Image Understanding*, 90(1):68–98.

[65] Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer.

[66] Kang, Z., Zhang, Z., Zhang, J., and Zlatanova, S. (2007). *Rapidly Realizing 3D Visualisation for Urban Street Based on Multi-Source Data Integration*, chapter Lecture Notes in Geoinformation and Cartography, pages 149–163. Springer Berlin Heidelberg.

[67] Klaus, A., Sormann, M., and Karner, K. (2006). Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 15–18, Washington, DC, USA. IEEE Computer Society.

[68] Kluckner, S., Pacher, G., Grabner, H., Bischof, H., and Bauer, J. (2007). A 3d teacher for car detection in aerial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8.

[69] Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized-via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

[70] Korc, F. and Forstner, W. (2008). Interpretation terrestrial images of urban scenes using discriminative random fields. In *Proceedings of the Congress of the International Society for Photogrammetry and Remote Sensing*, pages B3a: 291–296.

[71] Kou, Z. (2007). *Stacked Graphical Learning*. PhD thesis, Machine Learning Department, School of Computer Science, Carnegie Mellon University.

[72] Kumar, S. August, J. and Hebert, M. (2005a). Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In *Proceedings of the International Computer Vision and Pattern Recognition, EMMCVPR Workshop*.

[73] Kumar, S. (2005). *Models for Learning Spatial Interactions in Natural Images for Context-Based Classification*. PhD thesis, School of Computer Science, Carnegie Mellon University.

[74] Kumar, S. and Hebert, M. (2003a). Discriminative random fields: a discriminative framework for contextual interaction in classification. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1150–1157 vol.2.

[75] Kumar, S. and Hebert, M. (2003b). Man-made structure detection in natural images using a causal multiscale random field. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 119–126.

[76] Kumar, S. and Hebert, M. (2004). Discriminative fields for modeling spatial dependencies in natural images. In *Advances in Neural Information Processing Systems (NIPS)*.

[77] Kumar, S. and Hebert, M. (2005b). A hierarchical field framework for unified context-based classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1284–1291, Washington, DC, USA. IEEE Computer Society.

[78] Lafarge, F., Descombes, X., Zerubia, J., and Pierrot-Deseilligny, M. (2008). Automatic building extraction from DEMs using an object approach and application to the 3d-city modeling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(3):365–381.

[79] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Int. Conference on Machine Learning*.

[80] Lamdan, W. and Wolfson, H. (1988). Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the IEEE International Conference on Computer Vision*.

[81] Larlus, D. and Jurie, F. (2008). Combining appearance models and markov random fields for category level object segmentation. In *Proceedings of the International Computer Vision and Pattern Recognition*, pages 1–7.

[82] Lathrop, R. G., Styles, R. M., Seitzinger, S. P., and Bognar, J. A. (2001). Use of GIS mapping and modeling approaches to examine the spatial distribution of seagrasses in Barnegat bay, New Jersey. *Estuaries*, 24(6):904–916.

[83] Leberl, F., Bischof, H., Grabner, H., and Kluckner, S. (2007). Recognizing cars in aerial imagery to improve orthophotos. In *GIS '07: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pages 1–9, New York, USA. ACM.

[84] Leberl, F., Gruber, M., Ponticelli, M., Bernoegger, S., and Perko, R. (2003). The UltraCam large format aerial digital camera system. In *Proceedings of the ASPRS Annual Convention*. Anchorage USA. In CDROM.

[85] Leberl, F. and Szabo, J. (2005). Novel totally digital photogrammetric workflow. Technical report, Semana Geomatica, IGAC-Bogota, Colombia.

[86] Leberl, F. and Thurgood, J. (2004). The promise of softcopy photogrammetry revisited. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35 (Part B3):759–763.

[87] Lee, C. H., Greiner, R., and Schmidt, M. (2005). Support vector random fields for spatial classification. In *Proceedings of the Practice of Knowledge Discovery in Databases (PKDD)*.

[88] Lee, C.-H., Wang, S., Murtha, A., Brown, M. R., and Greiner, R. (2008). Segmenting brain tumors using pseudo—conditional random fields. In *Proceedings of the Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 359–366, Berlin, Heidelberg. Springer-Verlag.

[89] Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, Czech. Springer-Verlag.

[90] Lepetit, V. (2006). Keypoint recognition using randomized trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1465–1479. Member-Fua, Pascal.

[91] Levi, K. and Weiss, Y. (2004). Learning object detection from a small number of examples: The importanceof good features. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 53–60, Washington, DC, USA. IEEE Computer Society.

[92] Levin, A., Viola, P., and Freund, Y. (2003). Unsupervised improvement of visual detectors using co-training. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 626–633, Nice, France. IEEE Computer Society.

[93] Li, S. Z. (2001). *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[94] Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for object detection. In *Proceedings of the International Conference on Image Processing*, pages 900–903, New York, USA. IEEE.

[95] Lim, J. J., Arbelaez, P., Gu, C., and Malik, J. (2009). Context by region ancestry. In *Proceedings of the IEEE International Conference on Computer Vision*.

[96] Lin, C. and Nevatia, R. (1998). Building detection and description from a single intensity image. *Int. Journal Computer Vision and Image Understanding*, 72(2):101–121.

[97] Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395.

[98] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[99] Luo, T., Kramer, K., Samson, S., and Remsen, A. (2004). Active learning to recognize multiple types of plankton. *Proceedings of the Int. Conference on Pattern Recognition (ICPR)*, 3:478–481 Vol.3.

[100] Ma, X. (2008). *Learning coupled conditional random field for image decomposition : theory and application in object categorization*. PhD thesis, Massachusetts Institute of Technology.

[101] Ma, X. and Grimson, W. (2008). Learning coupled conditional random field for image decomposition with application on object categorization. *Proceedings of the Computer Vision and Pattern Recognition*, pages 1–8.

[102] Maree, R., Geurts, P., Piater, J., and Wehenkel, L. (2005). Random subwindows for robust image classification. In *Computer Vision and Pattern Recognition*, volume 1, pages 34–40 vol. 1.

[103] Markus, N. (2003). *Detection and reconstruction of buildings for automated map updating*. PhD thesis, Institut für Geodäsie und Photogrammetrie an der ETH, Zürich.

[104] Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. W. H. Freeman.

[105] Martinez, A. M. and Kak, A. C. (2001). PCA versus LDA. *Pattern Analysis and Machine Intelligence*, 23(2):228–233.

[106] Matei, B., Sawhney, H., Samarasekera, S., Kim, J., and Kumar, R. (2008). Building segmentation for densely built urban regions using aerial lidar data. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 1–8.

[107] Matikainen, L., Kaartinen, K., and Hyyppä (2007). Classification tree based building detection from laser scanner and aerial image data. In *Proceedings of ISPRS Workshop Laser Scanning*.

[108] Mayer, H. (1999). Automatic object extraction from aerial imagery—a survey focusing on buildings. *Computer Vision and Image Understanding*, 74(2):138–149.

[109] Mayer, H. (2008). Object extraction in photogrammetric computer vision. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):213–222.

[110] Mayer, H., Hinz, S., and Stilla, U. (2008). *Advances in Photogrammetry, Remote Sensing and Spatial Information Science*, chapter 16: Automated extraction of roads, buildings and vegetation from multi-source data, pages 213–226. ISPRS Congress book.

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

[112] Monteleoni, C. and Kaariainen, M. (2007). Practical online active learning for classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[113] Moon, H., Chellappa, R., and Rosenfeld, A. (2002). Performance analysis of a simple vehicle detection algorithm. *Image and Vision Computing*, 20(1):1–13.

[114] Moons, T., Frère, D., Vandekerckhove, J., and Gool, L. J. V. (1998). Automatic modeling and 3d reconstruction of urban house roofs from high resolution aerial imagery. In *Proceedings of the 5th European Conference on Computer Vision-Volume I*, pages 410–425, London, UK. Springer-Verlag.

[115] Mueller, S. and Zaum, D. W. (2005). Robust building detection in aerial images. In *ISPRS Workshop on Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms, and Evaluation (CMRT05)*.

[116] Murase, H. and Nayar, S. K. (1995). Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24.

[117] Murphy, K. P., Weiss, Y., and Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of Uncertainty in AI*, pages 467–475.

[118] Nagel, H.-H. (2004). Steps toward a cognitive vision system. *AI Magazine*, 25(2):31–50.

[119] Nair, V. and Clark, J. (2004). An unsupervised, online learning framework for moving object detection. In *Proceedings Conference on Computer Vision and Pattern Recognition*, volume 2, pages 317–324, Washington, DC, USA. IEEE.

[120] Nevatia, R., Lin, C., and Huertas, A. (1997). A system for building detection from aerial images. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images, Birkhaser Verlag*, pages 77–86.

[121] Nguyen, T. T., Grabner, H., Bischof, H., and Gruber, B. (2007). On-line boosting for car detection from aerial images. In *Proceedings of The IEEE International Conference on Computing and Telecommunication Technologies (RIVF)*, pages 87–95. IEEE.

[122] Nguyen, T. T., Nguyen, B. D., and Bischof, H. (2008). Efficient boosting-based active learning for specific object detection problems. In *Proceedings of The 5th IEEE International Conference on Computer Vision, Image and Signal Processing*.

[123] Nilsback, M. and Caputo, B. (2004). Cue integration through discriminative accumulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–578–II–585 Vol.2.

[124] Noronha, S. and Nevatia, R. (1997). Detection and description of buildings from multiple aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–594, Los Alamitos, CA, USA. IEEE Computer Society.

[125] Noronha, S. and Nevatia, R. (2001). Detection and modeling of buildings from multiple aerial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):501–518.

[126] Nowozin, S. and Lampert, C. H. (2009). Global connectivity potentials for random field models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[127] Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence*, 24(7):971–987.

[128] Olsen, B. and Knudsen, T. (2006). Automated change detection for validation and update of geodata. In *Proceedings of the 6th Geomatic Week*.

[129] Opelt, A., Pinz, A., Fussenegger, M., and Auer, P. (2006). Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):416–431.

[130] Osuna, E., Freund, R., and Girosi, F. (1997). Training support vector machines: an application to face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 130–136.

[131] Oza, N. and Russell, S. (2001a). Experimental comparisons of online and batch versions of bagging and boosting. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 359–364, San Francisco, CA, USA. ACM Press.

[132] Oza, N. and Russell, S. (2001b). Online bagging and boosting. In *Proceedings of the Artificial Intelligence and Statistics*, pages 105–112, Florida, USA. Morgan Kaufmann.

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

[134] Paparoditis, N., Cord, M., Jordan, M., and Cocquerez, J.-P. (1998). Building detection and reconstruction from mid-and high-resolution aerial imagery. *Computer Vision and Image Understanding*, 72(2):122–142.

[135] Paparoditis, N., Souchon, J., Martinoty, G., and Pierrot Deseilligny, M. (2006). High-end aerial digital cameras and their impact on the automation and quality of the production workflow. *Journal of Photogrammetry and Remote Sensing (IJPRS)*, 60(6):400–412.

[136] Park, J.-H. and Choi, Y.-K. (1996). On-line learning for active pattern recognition. *IEEE Signal Processing Letters*, 3(11):301–303.

[137] Persson, M. Sandvall, M. and Duckett, T. (2005). Automatic building detection from aerial images for mobile robot mapping. In *Symp. on Comp. Intel. in Robotics & Automation*.

[138] Pfeifer, N., Rutzinger, M., Rottensteiner, F., Muecke, W., and Hollaus, M. (2007). Extraction of building footprints from airborne laser scanning: Comparison and validation techniques. In *Urban Remote Sensing Joint Event*, pages 1–9.

[139] Pham, M.-T. and Cham, T.-J. (2007). Online learning asymmetric boosted classifiers for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[140] Pinto, N., Cox, D. D., and Dicarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1):e27+.

[141] Pinz, A. (2006). Object categorization. *Foundations and Trends in Computer Graphics and Vision*, 1(4):255–353.

[142] Plath, N., Toussaint, M., and Nakajima, S. (2009). Multi-class image segmentation using conditional random fields and global classification. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 817–824, New York, NY, USA. ACM.

[143] Porikli, F. (2005). Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 829–836, San Diego, CA, USA. IEEE Computer Society.

[144] Porway, J., Wang, K., Yao, B., and Zhu, S. C. (2008). A hierarchical and contextual model for aerial image understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[145] Quattoni, A., Collins, M., and Darrel, T. (2005). Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems (NIPS 2004)*, number 17.

[146] Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. (2007). Objects in context. In *ICCV*.

[147] Rajagopalan, A. N., Burlina, P., and Chellappa, R. (1999). Higher order statistical learning for vehicle detection in images. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1204–1209, Corfu, Greece. IEEE Computer Society.

[148] Remondino, F. and El-Hakim, S. (2006). Image-based 3d modelling: A review. *The Photogrammetric Record*, 21:269–291.

[149] Reynolds, J. and Murphy, K. (2007). Figure-ground segmentation using a hierarchical conditional random field. In *CRV '07: Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, pages 175–182, Washington, DC, USA. IEEE Computer Society.

[150] Reznik, S. and Mayer, H. (2007). Implicit shape models, model selection, and plane sweeping for 3d facade interpretation. In *Photogrammetric Image Analysis*, pages 173–178.

[151] Richard O. Duda, Peter E. Hart, D. G. S. (2001). *Pattern Classification*. New York: Wiley.

[152] Roberts, L. G. (1963). *Machine Perception of Three Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology.

[153] Roth, P., Grabner, H., Skočaj, D., Bischof, H., and Leonardis, A. (2005). On-line conservative learning for person detection. In *Proceedings Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 223–230, Beijing, China. IEEE.

[154] Roth, P. M. and Winter, M. (2008). Survey of appearance-based methods for object recognition. Technical report, TU Graz.

[155] Rottensteiner, F. (2008). Automated updating of building data bases from digital surface models and multi-spectral images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVII B3A:pp.265–270.

[156] Rudin, C., Daubechies, I., and Schapire, R. (2004). The dynamics of adaboost: Cyclic behavior and convergence of margins. *Journal of Machine Learning Research*, 5:1557–1595.

[157] Ruskone, R., Guigues, L., Airault, S., and Jamet, O. (1996). Vehicle detection on aerial images: A structural approach. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 900–904, Vienna, Austria. IEEE Computer Society.

[158] Santosh Kumar Divvala, Derek Hoiem, J. H. H. A. A. E. and Hebert, M. (2009). An empirical study of context in object detection. In *Proceedings of The Conference on Computer Vision and Pattern Recognition (CVPR)*.

[159] Schapire, R. (2003). The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification, Springer*.

[160] Schapire, R., Freund, Y., Bartlett, P., and Lee, W. (1997). Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings Interna-*

tional Conference on Machine Learning, pages 322–330, Nashville, TN, USA. Morgan Kaufmann.

[161] Schein, A. I. (2005). *Active learning for logistic regression*. PhD thesis, Philadelphia, PA, USA.

[162] Schlosser, C., Reitberger, J., and Hinz, S. (2003). Automatic car detection in high resolution urban scenes based on an adaptive 3D- model. In *Proceedings of GRSS/ISPRS Joint Workshop on Data Fusion and Remote Sensing over UrbanAreas, 2nd*, number 0-7803-7719-2, pages 167–171. IEEE.

[163] Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535.

[164] Schneiderman, H. and Kanade, T. (1998). Probabilistic modeling of local appearance and spatial relationships for object recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 45–51.

[165] Schneiderman, H. and Kanade, T. (2000). A statistical method for 3d object detection applied to faces and cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 746–751, Hilton Head, SC, USA. IEEE Computer Society.

[166] Schnitzspan, P., Fritz, M., and Schiele, B. (2008). Hierarchical support vector random fields: Joint training to combine local and global features. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 527–540, Berlin, Heidelberg. Springer-Verlag.

[167] Schroff, F., Criminisi, A., and Zisserman, A. (2008). Object class segmentation using random forests. In *Proceedings of the British Machine Vision Conference*.

[168] 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 *Proceedings of the European Conference on Computer Vision*, pages 1–15.

[169] Sirmacek, B. and Unsalan, C. (2008). Building detection from aerial images using invariant color features and shadow information. In *23rd Intl Symp. on ISCIS*, pages 1–5.

[170] Sivic, J., Russell, B., Efros, A. A., Zisserman, A., , and Freeman, B. (2005). Discovering objects and their location in images. In *Proceedings of the IEEE International Conference on Computer Vision*.

[171] Steinnocher, K. & Kressler, F. (2006). Change detection report. Technical report, EuroSDR.

[172] Stojmenovic, M. (2006). Real time machine learning based car detection in images with fast training. *Machine Vision and Applications*, 17(3):163 – 172.

[173] Stone, K. H. (1964). A guide to the interpretation and analysis of aerial photos. *Annals of the Association of American Geographers*, 54(3):318–328.

[174] Sung, K. and Niyogi, P. (1995). A formulation for active learning with applications to object detection. Technical Report AIM-1438.

[175] Sutton, C. and Mccallum, A. (2005). Piecewise training for undirected models. In *Neural Information Processing Systems*.

[176] Sutton, C., McCallum, A., and Rohanimanesh, K. (2007). Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal Machine Learning Research*, 8:693–723.

[177] Thompson, C. A., Califf, M. E., and Mooney, R. J. (1999). Active learning for natural language parsing and information extraction. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 406–414, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[178] Tieu, K. and Viola, P. (2000). Boosting image retrieval. In *Proceedings Conference on Computer Vision and Pattern Recognition*, pages 228–235, Hilton Head, SC, USA. IEEE Computer Society.

[179] Torralba, A., Murphy, K. P., and Freeman, W. T. (2005). Contextual models for object detection using boosted random fields. In *Neural Information Processing Systems*, pages 1401–1408.

[180] Toth, C. and Grejner-Brzezinska, D. (2005). Traffic flow estimation from airborne imaging sensors: A performance analysis. In *Proceedings of Workshop on High Resolution Earth Imaging and Geospatial Information*.

[181] Tsechpenakis, G., Wang, J., Mayer, B., and Metaxas, D. (2007). Coupling crfs and deformable models for 3d medical image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8.

[182] Tu, Z. (2008). Auto-context and its application to high-level vision tasks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[183] Turk, M. A. and Pentland, A. P. (1991). Face recognition using eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 586–591.

[184] Ullman, S. and Basri, R. (1991). Recognition by linear combinations of models. *Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006.

[185] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.

[186] Verbeek, J. and Triggs, B. (2007). Region classification with markov field aspect models. In *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pages 1–8.

[187] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–518.

[188] Vishwanathan, S. V. N., Schraudolph, N. N., Schmidt, M. W., and Murphy, K. P. (2006). Accelerated training of conditional random fields with stochastic gradient methods. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 969–976, New York, NY, USA. ACM.

[189] Wallach, H. (2002). Efficient training of conditional random fields. Master thesis, University of Edinburgh.

[190] Wang, Y. and Ji, Q. (2005). A dynamic conditional random field model for object segmentation in image sequences. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 264–270, Washington, DC, USA. IEEE Computer Society.

[191] Weber, M., Welling, M., and Perona, P. (2000). Towards automatic discovery of object categories. In *Proceedings of the IEEE Conference on Computer Vision Pattern Recognition*, pages 101–108.

[192] Werner, T. and Zisserman, A. (2002). New techniques for automated architectural reconstruction from photographs. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pages 541–555, London, UK. Springer-Verlag.

[193] Wolf, L. and Bileschi, S. (2006). A critical view of context. *Int. J. Comput. Vision*, 69(2):251–261.

[194] Wu, B., Ai, H., Huang, C., and Lao, S. (2004). Fast rotation invariant multi-view face detection based on real adaboost. In *Proceedings International Conference on Automatic Face and Gesture Recognition*, pages 79–84, Seoul, Korea. IEEE Computer Society.

[195] Wu, B. and Nevatia, R. (2007). Improving part based object detection by unsupervised, online boosting. In *Computer Vision and Pattern Recognition*, pages 1–8.

[196] Xie, M., Fu, K., and Wu, Y. (2006). Building recognition and reconstruction from aerial imagery and lidar data. In *Proceedings of the International Conference on Radar*, pages 1–4.

[197] Yao, J. and Zhang, Z. M. (2005). Semi-supervised learning based object detection in aerial imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1011–1016, Washington, DC, USA. IEEE Computer Society.

[198] Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2004). Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312.

[199] Zebedin, L., , Bauer, J., Karner, K., and Bischof, H. (2008). Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *Proceedings of the European Conference on Computer Vision*.

[200] Zebedin, L., Klaus, A., Gruber-Geymayer, B., and Karnera, K. (2006). Towards 3d map generation from digital aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(6):413–427.

[201] Zhang, H., Gao, W., Chen, Y., and Zhao, D. (2006). Object detection using spatial histogram features. *Image and Vision Computing*, 24(4):327–341.

[202] Zhao, T. and Nevatia, R. (2003). Car detection in low resolution images. *Image and Vision Computing*, 21(8):693 – 703.

[203] Zhong, P. and Wang, R. (2006). Object detection based on combination of conditional random field and markov random field. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 160–163.

[204] Zhou, Q.-Y. and Neumann, U. (2009). A streaming framework for seamless building reconstruction from large-scale aerial lidar data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[205] Zhu, Q., Hu, M., Zhang, Y., and Du, Z. (2009). Research and practice in three-dimensional city modeling. *Geo-Spatial Information Science*, 12(1):18–24.

[206] Zimmermann, P. (2000). A new framework for automatic building detection analyzing multiple cue data. *International Archives of Photogrammetry and Remote Sensing*, 33:1063–1070.