DISS. ETH NO. 21923

High-Resolution 3D Layout from a Single View

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

MUHAMMAD ZEESHAN ZIA

Master of Science (M.Sc.), Technische Universität München

born on 26.07.1984

citizen of Pakistan

accepted on the recommendation of

Prof. Dr. Konrad Schindler, ETH Zurich
Prof. Dr. Tinne Tuytelaars, KU Leuven

2014

IGP Mitteilungen Nr. 114
High-Resolution 3D Layout from a Single View
Muhammad Zeeshan Zia

Published by:
Institute of Geodesy and Photogrammetry
ETH ZURICH
CH-8093 Zurich

ISBN 978-3-03837-000-0

Abstract

Scene understanding based on photographic images has been the holy grail of computer vision ever since the field came into existence some 50 years ago. Since computer vision comes from an Artificial Intelligence background, it is no surprise that most early efforts were directed at fine-grained interpretation of the underlying scene from image data. Unfortunately, the attempts proved far ahead of their time and were unsuccessful in tackling real-world noise and clutter, due to unavailability of vital building blocks that came into existence only decades later as well as severely limited computational resources.

In this thesis, we consider the problem of detailed 3D scene level reasoning from a single view image in the light of modern developments in vision and adjoining fields. Bottom-up scene understanding relies on object detections, but unfortunately the hypotheses provided by most current object models are in the form of coarse 2D or 3D bounding boxes, which provide very little geometric information - not enough to model fine-grained interactions between object instances. On the other hand, a number of detailed 3D representations of object geometry were proposed in the early days of computer vision, which provided rich description of the modeled objects. At the time, they proved difficult to match robustly to real world images. However over the past decade or so, developments in local image descriptors, discriminative classification, and numerical optimization methods have made it possible to revive such approaches for 3D reasoning and apply them to challenging real-world images. Thus we revisit detailed 3D representations for object classes, and apply them to the task of scene-level reasoning. The motivation also comes from recent revival of coarse grained 3D modeling for scene understanding, and demonstrations of its effectiveness for 3D interpretation as well as 2D recognition. These successes raise the question of whether finer-grained 3D modeling could further aid scene-level understanding, which we try to answer in our work.

We start with 3D CAD training data to learn detailed 3D object class representations, which can estimate 3D object geometry from a single image. We demonstrate applying this representation for accurate estimation of object shape, as well as for novel applications namely, ultra-wide baseline matching and fine-grained object categorization. Next, we add an occluder representation comprising of a set of occluder masks, which enables the detailed 3D object model to be applied to occluded object instances, demonstrated over a dataset with severely occluded objects. This object representation is lifted to metric 3D space, and we jointly model multiple object instances in a common frame. Object interactions are modeled at the high-resolution of 3D wireframe vertices: deterministically modeling object-object occlusions and long-range dependencies enforcing all objects to lie on a common ground plane, both of which stabilize 3D estimation. Here, we demonstrate precise metric 3D reconstruction of scene layout on a challenging street scenes dataset. We evaluate parts of our approach on five different datasets in total, and demonstrate superior performance to state-of-the-art over different measures of detection quality. Overall, the results support that detailed 3D reasoning benefits both at the level of individual objects, and at the level of entire scenes.

Zusammenfassung

Seit sich die *computer vision* vor ca. 50 Jahren als eigenständiges Feld etabliert hat ist das Szenenverstehen, also die semantische Interpretation der abgebildeten Szene, eines ihrer fundamentalen Probleme. Da der Ursprung der *computer vision* in der künstlichen Intelligenz liegt überrascht es nicht, dass zu ihren Zielen von Beginn an das automatische Verstehen der beobachteten Szene gehörte. Aus heutiger Sicht ist es verwundert es auch nicht, dass die anfänglichen Versuche scheiterten, einerseits weil wesentliche Grundlagen erst Jahrzehnte später entwickelt wurden, andererseits weil die damaligen Computer nicht die notwendige Rechenleistung hatten.

Die vorliegende Arbeit untersucht das Problem des detaillierten, 3-dimensionalen Szenenverstehens auf Basis eines Einzelbildes, ausgehend von den heutigen Möglichkeiten der *computer vision* und verwandter Disziplinen. Ein grundlegender Baustein des Szenenverstehens ist die Erkennung von Objekten im Bild. Die gebräuchlichen Detektoren liefern jedoch als Objektmodell nur 2D oder 3D *bounding boxes*, und diese grobe Repräsentation ist nicht geeignet, die Objektgeometrie und die Interaktionen zwischen verschiedenen Objekten im Detail zu modellieren. In Gegensatz dazu wurden in der Frühzeit der *computer vision* Repräsentationen der Objektgeometrie entwickelt, die eine wesentlich höheren Detailgrad aufweisen. Es gelang damals aber nicht zuverlässig, das Modell mit dem Bildinhalt in Korrespondenz zu bringen. Die Entwicklungen der letzten Jahre im Bereich der lokalen Bild-Deskriptoren, der diskriminativen Klassifikation und der numerischen Optimierung ermöglichen es, diese Ansätze wiederzubeleben und auf das Verstehen komplexer 3-dimensionaler Szene anzuwenden. In der vorliegenden Arbeit wird daher eine solches klassisches, detailreiches 3D Objektmodell für das bildbasierte Szenenverstehen benutzt. Der vorgestellte Ansatz ist unter anderem dadurch motiviert, dass in den letzten Jahren das 3-dimensionale Szenenverstehen – mit eher groben Modellen – wieder vermehrt untersucht wurde. Dabei zeigte sich ,dass es sowohl für die 3D Modellierung als auch für die Objekterkennung im Bild Vorteile bringt. Diese Erfolge werfen die Frage auf, ob detailliertere Modelle das Szenenverstehen weiter verbessern können. Die vorliegende Arbeit ist ein Versuch, die Frage zu beantworten.

Den Ausgangspunkt der Arbeit bilden 3D CAD-Modelle. Auf deren Basis werden detaillierte, deformierbare Objektrepräsentationen gelernt, mit deren Hilfe die 3D Geometrie des Objekts auf Basis eines Einzelbildes geschätzt werden kann. Neben der Rekonstruktion der genauen geometrischen Objektform ermöglichen solche Modelle auch neue Anwendungen wie das *matching* über extrem grosse Basislinien und die Klassifizierung in nur durch geometrische Details unterscheidbare Unterkategorien. Um Verdeckungen in den Bildern verarbeiten zu können wird das Modell um eine Verdeckungsmaske erweitert. Die Maske ermöglicht es, die Verdeckung einzelner Objektteile darzustellen, und es wird gezeigt, dass sich damit auch stark verdeckte Objektinstanzen detektieren lassen. Schliesslich wird das Modell noch so modifiziert, dass Objekte im metrischen 3D Koordinaten repräsentiert werden. Somit können mehrere Objekte in einem gemeinsamen Koordinatensystem modelliert werden. Weiters werden Interaktionen zwischen den verschiedenen Objekten auf dem Niveau einzelner Objektpunkte und -flächen berücksichtigt,

im speziellen gegenseitige Verdeckungen und eine gemeinsame Geländeebene, auf der alle Objekte stehen. Es wird gezeigt, dass mit einem derart stabilisierten Modell komplexe Strassenszenen metrisch korrekt rekonstruiert werden können. Die einzelnen Teile der vorgeschlagenen Methode wurden auf mehreren verschiedenen Datensätzen evaluiert, dabei wurden signifikante Verbesserungen hinsichtlicher verschiedener Qualitätsmasse beobachtet. Insgesamt stützen die Ergebnisse die Hypothese, dass die detaillierte 3-dimensionale Modellierung vorteilhaft für das Bildverstehen ist, sowohl auf der Stufe einzelner Objekte als auch auf der Stufe kompletter Szenen.

Acknowledgements

First and foremost, I want to thank **Prof. Konrad Schindler** for giving me the opportunity to pursue a PhD in his supervision. He is the best supervisor one can ask for, possessing incredible scientific knowledge and reasoning skills, very open-minded, having strong management abilities, and always sympathetic and polite to his students. In science, I am particularly impressed by (and try to copy) his way of putting things in context, and reducing the heap of technical details to a few key ideas. His deep insights in so many diverse tools and problems in computer vision can only be described as inspirational. He treats his PhD students as equals from the beginning of their studies, and gives them a decent amount of freedom with proposing and implementing their own ideas, which is a unique characteristic of his supervision. He was also very supportive w.r.t. my wife's visa issues, my internship at Qualcomm, and the multiple customized recommendation letters that he wrote for me. Finally his thorough review and corrections on my dissertation were very helpful in bringing this thesis to its present form.

I also had the good fortune of collaborating with **Dr. Michael Stark** from the beginning of this project. In fact, this thesis is built on top of the code base that he developed for Stark et al. (2010). He made valuable contributions w.r.t. ideas as well as writing up of our papers. His pursuit of excellence regarding experimentation and presentation pushed me to improve my output, and made me more disciplined. He also wrote multiple recommendation letters for me. I am very thankful to him for his mentorship.

Prof. Bernt Schiele collaborated on my PhD project during its first year, while we were part of his chair at TU Darmstadt. I learned to be thorough in doing experiments, and visualizing intermediate data to develop insights into the behaviour of vision algorithms, from him. Also, listening to his comments at the prep talks of his students for various conference and thesis presentations and at our biannual retreats, honed my presentation and scientific reasoning skills. I am grateful for his support.

I am also thankful to **Prof. Tinne Tuytelaars** for agreeing to be my examiner, and for contributing her time to examining this thesis. It will be a big honour for me to have her name on my thesis and CV.

Another important contribution to this thesis, is by **Ms. Monique Berger Lande** whose Swiss-quality administration enabled me to peacefully stay hidden in my office and focus on my research. She took care of all the real-world bureaucratic hurdles, at times even personal issues such as talking to visa officials and landlords, and from organizing a new contract for me when I went abroad for internship, to taking care of retreats, conferences and summer school trips – for which I am very thankful. Seriously, I have already started missing her support in the new city!

Javier Montoya and **Christoph Vogel** have been good for my morale, being usually at the office at odd hours like me. I have had hundreds of chats with them when the office was otherwise empty. I learned to be more passionate towards people and to try having a social component in my life from Javier, and to be mathematically rigorous and helpful to students from Christoph. Also many thanks to both of them for translating many of my

official correspondences including the abstract for this thesis to German language.

Also the friendly and welcoming environment at the lab couldn't be possible without **Nusret**, **Haris**, **Jan**, **Jiaojiao**, **Wilfried**, **Michal**, **Silvano**, **Maros**, **Manos**, **Stefan**, **Sultan**, **Piotr**, **Charis**, **Pascal**, **David**, **Fee**, **Nora**, **Lisa**, **Ines**, and **Kirushiga**. My helpful colleagues and friends in Darmstadt where I originally started this PhD include **Ulf**, **Sandra**, **Anton**, **Diane**, **Paul**, **Christian**, **Micha**, **Kristof**, **Nico**, **Marcus**, **Leonid**, **Faraz**, and **Asad**. I also enjoyed many interesting discussions with the academic guests at the group, specially **Prof. David Suter** and **Dr. Alexander Velizhev**, which introduced me to foreign academic cultures. My stay in Zurich was made a lot pleasant due to my friendship with **Farrukh**, **Qasim**, **Asim**, **Afzal**, **Najeeb**, **Shabir**, **Sarfaraz**, and **Sarmad**. Further, **Owais** proof-read an early version of this thesis. I am grateful to all these people for their social support as well as for teaching me something about life in their own way.

Last but not the least, my **parents** and **wife** deserve a lot of praise for their support. My parents supported my lavish extra-curricular activities at a time when these didn't seem to carry any career benefits. My wife tolerated my awful office timings, my ill-planed internship, and now our move to an unknown and tougher city. Thank you very much.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans are able to infer a lot of detail about the underlying scene from a picture. They know what objects are present, their poses, the 3D spatial layout of different scene elements (compact, well-delineated objects and "stuff" such as walls, sky, ground), as well as how these elements interact with each other. Human visual processing system not only recognizes objects in isolation but also considers context: the recognition of each scene component improves the recognition of other components. A yellow round blob in the hands of a player standing in a tennis court, would immediately be recognized as a tennis ball. However, exactly the same pixels cut out from the yellow blob shown on a dining table with some vegetables, would be "recognized" as a lemon.

Since its beginning, detailed scene understanding has been the holy grail of computer vision research. Researchers proposed a number of rich representations for objects and scenes and applied them to images acquired under controlled settings, such as on blocks arranged in different configurations pictured against a clean background. We discuss in detail, such legacy systems for 3D object recognition (Section 1.3.4) and scene-level reasoning (Section 1.3.6), and mention several others in the later chapters (Chapter 3, 4, 5). Unfortunately, these algorithms could not work with real-world images due to appearance variations arising from occlusions, viewpoint, intra-class shape, lighting, and texture, as well as due to distractions arising from background clutter. These robustness issues were caused by a deterministic approach to vision problems that could not reliably take various distractions found in image data into account. Furthermore, the algorithms that search for best hypothesis among a large or even infinite set of hypotheses (optimization algorithms) were severely limited by computational resources.

Due to these nuisances, subsequent research traded off modeling accuracy for robustness in matching, *e.g.* by representing objects by the statistics of local features in an image window. Consequently there have been significant advances in local shape features, discriminative classifiers, and efficient techniques of approximate probabilistic inference. This has led to impressive performance on recognition of a variety of object classes, on region labelling, and on scene classification problems over the last decade, but the extent to which interactions between scene entities can be modeled is still fairly limited. The objective of this thesis is to revisit ideas from the very early days of computer vision in the light of modern machine learning and optimization techniques, while benefitting from superior computational resources.

Overriding research questions. This thesis looks at whether recent developments have taken us to a point where we may use detailed 3D object models, like those explored in the early days of computer vision. In this context it asks the following questions:

1. Can such models be made to work on challenging real-world imagery as opposed to images taken under laboratory settings? Is it beneficial to use fine-grained 3D object models as opposed to models that simply output 2D bounding boxes around objects?
2. Since one major issue in realistic scenes is partial occlusion: how can such models be made to cope with occlusion?
3. Are there any advantages in applying detailed 3D object representations to the task of high resolution object interaction modeling in 3D space, over coarser modeling approaches?

1.1 Motivation

As mentioned already, the past decade saw rapid advances in object recognition technology fueled by sustained interest of a large part of the computer vision community. Independent object recognition (*i.e.* without considering any context) remained the area with the greatest number of paper submissions and acceptances at flagship computer vision conferences like CVPR for many years. However recognition performance of independent 2D appearance representations started to saturate (*e.g.* $\approx$35$\%$ average precision for the well-known Pascal VOC challenge Everingham et al., 2010). Although *per se* this does not mean that more complicated models are the way to go, it does still raise the question whether a 3D representation, which allows top-down segmentation, reconstruction, and recognition in a more integrated way would alleviate some of the difficulties.

More recently (about the time this thesis started), researchers had revived coarse 3D geometric representations in the context of indoor and outdoor scene understanding (Section 1.3.5). They demonstrated the benefits of 3D geometric reasoning not only w.r.t. greater expressiveness of the models but also increased 2D recognition performance. This hinted that even richer 3D representations amenable to joint reasoning about multiple scene element could be further beneficial for scene understanding.

These insights led us to pursue a course of research where we developed a detailed 3D object model, exploring the benefits offered by such models, and gradually extending it for reasoning about interactions first in the image space and then in metric 3D space.

1.2 Methodology and Contributions

The work done in this thesis has been successively disseminated across several papers. Since this thesis is being presented as a *paper dissertation* (in compliance with ETH-Zurich Doctorate Ordinance of 2008), we include the key papers as Chapters 3, 4, and 5.

Approach. We follow a step-by-step approach to building a 3D scene-level reasoning system. To this end, we start by developing a detailed 3D object model trained on 3D CAD exemplars. This representation provides us with a 3D shape hypothesis for an isolated fully visible object under challenging viewing conditions (variations in azimuth, elevation, scale, background, and lighting) from a single view image. In fact, the original model that we introduce in Zia et al. (2011) requires huge computational and memory resources, since the appearance representation requires a separate AdaBoost classifier for each part and viewpoint. This means training and then evaluating more than 5000 classifiers in a sliding-window fashion on the test image (36 parts x 144 viewpoints = 5184 classifiers). This prohibitively expensive requirement is lifted in the follow-up work (Chapter 3; Zia et al., 2013) by introducing a viewpoint-invariant Random Forest classifier. We perform extensive experimentation on two standard datasets and show that such expressiveness in modeling is beneficial even without performing higher scene-level reasoning: for continuous viewpoint estimation, ultra-wide baseline matching, and even fine-grained categorization.

The next step is adding an explicit occluder representation to the model (Chapter 4), so that even partially occluded objects may be reliably detected and reconstructed. As no dataset with well-labeled instances of severely occluded objects was available, we created our own dataset for evaluation, and made it publicly available. Specifically we collected and labeled a new test set comprising of 100 challenging street images from around Zurich, biased towards challenging occluded cases. The experiments on this test set indicate the efficacy of our explicit occlusion modeling.

Next we lift the object model from image space to metric 3D space (Chapter 5), by training the detailed 3D geometric representation on 3D CAD models, which have been scaled according to their real-world dimensions. Next, we perform an intermediate 2D-to-3D lifting step by coarse grid search. This inherently converts our inference to a 3D scene reconstruction procedure, iteratively verifying the reconstruction against image evidence. We further introduce two modes of explicitly modeling object-object interactions: a common ground plane and determinstic object-object occlusion reasoning. We also make necessary modifications to our inference procedure to cope with this increase in search space dimensionality. We evaluate the approach on the newly introduced *KITTI* dataset of Geiger et al. (2012) which comes with camera calibration and 3D object labels[1]. The detailed experimentation for 3D object localization, 3D viewpoint estimation, and occlusion prediction indicate that such higher fine-grained scene-level reasoning indeed improves accuracy over all these measures.

The evaluations performed during these three stages of the thesis, effectively answer the original research questions that we set out to answer (see *Overriding research questions*).

Contributions in Chapter 3. The contributions made in the development of the detailed 3D geometric model are as follows:

1. We revisit a detailed 3D geometric model in the light of modern advances in representation, learning, and optimization. We show that for certain object types classical

[1]The KITTI dataset was not available for the earlier chapters.

3D geometric object class representations can deliver object hypotheses with far more geometric detail than current detectors.

2. We demonstrate that a 3D object model enriched with local appearance descriptors, and coupled with a discriminative classifier allows accurate determination of object shape and continuous pose. In particular, our model outperforms state-of-the-art techniques for object viewpoint estimation over two standard multi view datasets.

3. We successfully show the benefit of detailed 3D reasoning for two novel applications: *(i)* a geometric modeling task namely ultra-wide baseline matching, where we recover relative camera poses over baselines up to 180° apart again improving over state-of-the-art by large margins, and (ii) predicting fine-grained object categories (different types of cars and bicycles) based on our wireframe estimates with encouraging results.

Contributions in Chapter 4. The contributions toward occlusion modeling in detailed 3D object representations are described below:

1. We propose a complete framework for detection and reconstruction of severely occluded object in monocular images, starting with a variant of the *poselets* idea (Bourdev and Malik, 2009) adapted to the needs of our 3D object model.

2. Alongside local part detectors, our appearance model now integrates evidence from the configuration detectors which profit from considering a relatively larger spatial window in predicting the local part location. These predictions are valuable because separate configurations are learned for separate viewpoints and major shape variations, causing the part locations to be well correlated.

3. An explicit occlusion model represented by a set of occluder masks and a neighborhood definition among them that allows for efficient sampling of the masks.

4. We experimentally demonstrate the efficacy of our approach on strongly occluded objects, in situations where representations without an occlusion model fail.

Contributions in Chapter 5. The contributions made while attempting to jointly model multiple object instances in a common 3D frame can be described as follows:

1. To the best of our knowledge, our work is the first attempt at exploring both short range and long range object-object interactions within a scene at high geometric resolution (individiual vertices of a wireframe model).

2. We further capitalize on our detailed object model with explicit 3D pose and explicit parts for occlusion modeling, by integrating deterministic reasoning about occlusion among detected objects with a generative probabilistic model of unknown occluders (Chapter 4). This again yields better 3D localization accuracies as compared to indepedently estimating occlusion for each individual object.

3. Finally, we present a detailed experimental evaluation of the 3D scene model on the *KITTI* street scene dataset (Geiger et al., 2012), and demonstrate the ability to localize 44% of highly occluded cars accurately with an accuracy of 1 meter.

We defer a much more detailed description of methodology, contributions, and experimental evaluation to Chapters 3, 4, and 5. While discussing contributions, we should mention that we have already made publicly available all the image sets (training and testing), as well as annotations ($\approx$2500 training and testing images labeled at high resolution). Also most of the trained models, and source code developed during this thesis has already been made publicly available, and the remaining portions will be published in due course.

1.2.1 Overview of the thesis

We start by discussing key ideas from literature as well as highlighting the contributions made in this thesis in the current chapter (Chapter 1).

Chapter 2 introduces some standard machinery from computer vision and machine learning for completeness. We explain the fundamental principles behind the building blocks in our system.

The next three chapters successively build a 3D scene-level reasoning system from the ground up. The first stage in the development is a detailed 3D geometric object model which can deal with multiple views, however can only reason about fully visible objects in isolation, discussed in Chapter 3. Apart from discussing approach and experiments, we provide a more detailed literature survey on object modeling.

Chapter 4 augments our object model with an explicit occlusion representation. We present a complete system for recognition and reconstruction given a single view image, from obtaining coarse 2D bounding box level detections to refining them into fine-grained 3D shape hypotheses. However, this chapter still deals only with individual object instances. The literature survey includes a more thorough overview of recent occlusion modeling work. We present experiments on a street scenes data set that we collected from around Zurich, comprising of severely occluded cars.

In Chapter 5, we discuss joint modeling of multiple objects in a common 3D coordinate frame, invoking object-object interactions at a high geometric resolution. 3D object locations as well as qualitative 3D reconstructions are obtained for the challenging KITTI (Geiger et al., 2012) dataset, from single view images. We demonstrate superior results for 3D localization and viewpoint estimation accuracy as compared to baselines which do not utilize joint object modeling.

Chapter 6 concludes the thesis. After summarizing the results from the previous chapters, it discusses the lessons learned in the course of the thesis, and critically analyzes the proposed system. It also proposes future directions, both at the level of technical upgrades and suggestions towards the advancement of scene understanding systems from a broader perspective.

1.3 Related work

The work done in this thesis touches upon various problems in computer vision: object class recognition, viewpoint invariance, occlusion modeling, scene-level understanding, even ultra-wide baseline matching, part appearance sharing, and fine-grained categorization. We introduce some of the key ideas explored in the literature in the following sections, and defer a more detailed listing of important work to later chapters.

1.3.1 Object class detection

Research in object detection is divided into two categories, namely, specific object detection and object class detection. Specific object detection deals with recognizing a particular object exemplar, *e.g.* the book "Computer Vision" by Forsyth and Ponce, as opposed to the generic object class "book" (dealt with in object class detection). While specific object detection is challenging in itself, it does not have to deal with the added problem of being invariant to intra-class appearance and shape variations *e.g.* between a sedan and a station wagon. Dominant approaches to object class detection include: (1) bag of words models, (2) approaches based on the generalized Hough transform, and (3) sliding-window models (rigid and part-based).

Bag of Words models. "Bag of words" (*BoW*) approaches for visual object class recognition, introduced by Sivic and Zisserman (2003) and Csurka et al. (2004), represent an object class as a collection of "visual words" (usually *local invariant features*, surveyed in Tuytelaars and Mikolajczyk (2008)) discarding the relative locations of these words. *Local invariant features* are points or regions in the scene, which can be accurately detected when pictured from different illumination, viewpoints, and distance. These approaches quantize the local invariant features extracted from many training exemplars to a fixed visual vocabulary. Thus for "face" class the visual words might comprise of nose, ears, eyes, and lips, but discard the fact that eyes lie side-by-side, and are above the nose which is above the lip. Discarding the geometry obviously makes the model invariant to intra-class shape variations, however not utilizing such an informative cue does not intuitively seem like it would yield strong detectors. Surprisingly, BoW approaches remained the most powerful methods for 2D object class recognition over the past decade 2000-2010 (Everingham et al., 2010). However over the last few years, part-based methods have matured *e.g.* Felzenszwalb et al. (2010), and outperform *BoW* models.

Generalized Hough Voting. Generalized Hough Voting methods augment *BoW*-style models with implicit shape information which helps group together visual word detections that are likely to belong to the same object instance. This is achieved by letting the visual word activations in a test image cast votes for object location and scale. Bounding box level detections are then obtained by finding modes in the voting space. An important approach which falls in this category is the *Implicit Shape Model (ISM)* of Leibe et al. (2006). Given a labeled training set comprising of 2D bounding boxes around exemplars of the object class of interest, *ISM* computes local interest points on the test images, maintaining the displacement to the object center and relative scale for each interest

point. These interest points are then clustered on the basis of appearance, and form a "codebook" of visual words. For detecting objects in an unseen image, the interest point detections are matched against the codebook, and probabilistic (soft) votes are cast according to the stored spatial distribution of the matched codebook entry. A standard "scale-adaptive" mode estimation scheme is used to find the maxima in the voting space. In practice, they further use a verification stage against stored segmentation masks to filter out erroneous votes and improve the detection scores. Perhaps the most successful variant which mixes Hough voting with sliding-window classification is the *Poselets* method of Bourdev and Malik (2009) (described in Section 1.3.3).

As opposed to *BoW* detectors, this approach models the relative positions of the visual words, however not as rigidly as the part-based models (discussed next). Also this method is naturally suited to handle partial occlusions, since even if some of the interest points on an object are occluded, the visible ones can still vote for the correct object center. However, this method like the *BoW* model, suffers from relatively high false-positive rates since cluttered backgrounds may give rise to many similar looking local features. Further, lately it has become established that dense features coupled with discriminative techniques outperform sparse interest point based approaches for rigid object class detection. These ideas have been integrated into the framework of Generalized Hough Voting by Gall and Lempitsky (2009) who utilize a Random Forest (Section 2.2.2) to perform the voting instead of forming an explicit codebook. However, dense feature coverage and discriminative methods (classification, as opposed to regression used here) has been more successful and explored far more in the context of sliding-window framework, which we cover in the next few paragraphs.

Sliding window detectors: Rigid models. Another dominant object class detection approach, *sliding-window based detectors*, evaluate a "sliding" window over the entire test image (since the object of interest can be present at any location in the image), at a range of scales (since the object can be of different sizes in the image). Such approaches can be further divided into two sub-categories: rigid models and part-based models. Rigid sliding window detectors effectively reduce the problem of object detection to binary classification (object *vs.* background) combined with exhaustive search. One important example of a rigid sliding window detector is the *Histogram of Oriented Gradients (HOG)* model of Dalal and Triggs (2005). This method is based on computing local image gradients. *Image gradient* at a certain pixel location for a certain direction, is the magnitude of change in image intensity in that direction. In this approach, the detector window is divided into small rectangular cells, and for each cell a histogram of gradient orientations is computed. Each bin in the histogram corresponds to a range of gradient orientations (*e.g.* 0°–180° divided evenly in 9 bins), and is filled with the sum of per-pixel gradient magnitudes. The overall window histogram comprises of a 1D vector appending all the component histograms. Assignment of whether a window contains an instance of the object class of interest or not is carried out by passing the histogram as input to a *classifier*. A *classifier* is an algorithm which outputs one of a discrete set of labels given an input vector (Section 2.2). Before the system can be applied to unseen (test) images, the classifier (specifically a Support Vector Machines or SVM classifier) "learns" the values of a set of tuning parameters (SVM weights), over a set of labeled input vectors,

in this case, labeled into object and non-object classes. It should be noted that the model absorbs intra-class variations by local spatial and orientation binning, since such binning allows minor local geometric and photometric variations. Although many variants of gradient histograms had been proposed in earlier work, Dalal and Triggs (2005) pursued a detailed analysis of various parameters involved in designing such histograms, and reported a number of lessons learnt. These suggestion for robust design of rigid models include that spatial sampling should be kept coarse, the orientation sampling should be fine, and that local constrast normalization (after combining multiple adjacent cells into blocks) should always be performed. This model while fairly robust, however is of little use to us directly since the global template has no “parts” and is not viewpoint-invariant.

Sliding window detectors: Part-based models. Part-based models search for parts of the object in each test window, modeling the relative part displacements in one way or the other. While intuitively it makes sense to incorporate this additional information about relative part locations (in addition to part appearances), it took a long time before a part-based model could really perform better in practice than the *BoW* models (Felzenszwalb et al., 2010). One way of modeling the relative locations of the parts was introduced in the framework called “Pictorial Structures" (Fischler and Elschlager, 1973). The Pictorial Structures framework penalizes deviations of part locations from their mean positions relative to a “root” part, on overall object detection score (penalties subtracted from the sum of part appearance scores). This can be thought of as springs attaching the root part (*e.g.* the nose part in a face model) to the other parts (eyes, ears, lips), the springs being at rest when the parts are at their mean position. It takes effort proportional to the amount of displacement, to stretch a spring and bring a part to any other location, *e.g.* to bring the nose above the eyes. However a small deformation may be allowable if it significantly increases the part appearance score. The object detection process attempts to maximize the overall object detection score by displacing the parts, in the test window. The most important variation of this idea, which is the top performer among the available and established “standard” methods on the Pascal VOC Challenge and has become a standard “go to” detector in the field, is the *Deformable Parts Model (DPM)* of Felzenszwalb et al. (2010). In this model, the root part comprises of a rigid HOG template, and the parts are also smaller HOG templates, all learnt on training data. The model parameters are automatically learnt over labeled training images (2D bounding boxes around example objects). However, the number of parts is specified manually. Relatively large intra-class variations are accounted for by these deformations, whereas local variations are dealt with by the spatial and orientation binning in the HOG templates. Dynamic programming is used to efficiently search the space of part location hypotheses when matching. The approach leverages on advances in discriminative learning techniques as well as a number of algorithmic tricks (*e.g.* to deal with large amounts of training data).

We use a bank of *DPM*s as the first step in our processing pipeline (Chapter 4), *i.e.* to detect candidate 2D object locations, which we then refine using our detailed 3D object model. Our detailed object model is also part-based, however we utilize an Active Shape Model (ASM) formulation (Section 2.3) to model relative part locations. The advantage of this formulation is that it constrains all relative part locations to always output an over-

all plausible object geometry (as seen in the training data), which is not the case with a star-shaped model (all parts connected only loosely to a root part). However, the disadvantage of our approach is that the optimization is more expensive, causing us to resort to simulation-based approaches (Section 2.4).

1.3.2 Multi view recognition

Objects can have significantly different appearance when viewed from different poses, *e.g.* consider the side-view *vs.* front-view of bicycles, or the top view *vs.* front view of airplanes. Besides being tolerant to intra-class variations, object class detectors also need to model appearance variations across viewpoints. The most common approach for achieving this is to train a "bank" of several independent single view detectors, and combine their detections by some arbitration function. One example is the *DPM-3D-Constraints* approach of Pepik et al. (2013), which defines parts to be cubical volumes on a set of aligned 3D CAD models of the object class of interest. The approach learns many single view DPM detectors on 2D projections of these CAD models. The additional benefit of keeping parts consistent across viewpoint detectors is that they can establish top-down correspondences among multi view test images, which can be useful for higher-level reasoning. However to get good results they need to densely model the relevant region of the viewing sphere, resulting in too many independent detectors which have to be evaluated over the same test image - and thus very high computational load. On the other hand, if the expected set of views is not densely modeled, object instances with an unseen pose may be missed. Thus some multi view detectors employ different degrees of coupling between different views in their representation.

One such detector, Thomas et al. (2006) augments the Implicit Shape Model (discussed in Section 1.3.1, *Generalized Hough Voting*), to "transfer" votes across ISM codebooks trained for different views through *activation links*. For training, it requires images taken from different views for each specific object instance in the set, and estimates "region-tracks" across views for each specific object. The region-tracks are correspondences between local interest regions found across different views for the objects. A separate *ISM* model is trained for each viewpoint using training examples only for that viewpoint. The region-tracks are then used to establish linkages (per object instance) between entries of the ISM codebooks called *activation links*. To detect objects in a test image, the approach first evaluates the bank of viewpoint-dependent ISM models separately on the image, and agglomerates the predicted 2D bounding boxes in to clusters (w.r.t. location and scale), discarding the clusters that have little support. Finally, it performs vote transfer among the codebooks for the detections in each agglomerated cluster. The advantage of this approach is that if the pose of a test object falls in between two viewpoint-dependent ISMs, evidence from both the views can be corroborated resulting in a more confident detection hypothesis, while requiring fewer independent ISMs.

Similar to Thomas et al. (2006), we also perform viewpoint estimation in two stages, first is a coarse stage comprising of a bank of single view detectors which provides us with a 2D object bounding box and a discrete 2D viewpoint (classified into one of eight classes). The second stage refines the viewpoint estimate by projecting a 3D wireframe model into the test image, searching in continuous viewpoint space. Utilizing a true 3D model allows

us to achieve highly accurate estimates of viewpoint (Chapter 3), which are valuable at different levels: from low-level tasks like occlusion and space occupancy reasoning, to higher-level ones such as enforcing long-range regularities (*e.g.* cars are parked usually parallel to each other).

1.3.3 Occlusion invariance

A major challenge for visual object detection is missing image evidence due to partial occlusion, and while *BoW* and *ISM*-style detectors are naturally suited to dealing with occlusions, they are nevertheless not competitive for object classes with low or moderate articulation. Thus researchers have been pursuing occlusion modeling in the context of rigid and part-based models. In the absence of an occlusion model, sliding-window approaches cannot distinguish between occluded and unoccluded portions of the object, causing the estimates to be inaccurate. One representative approach that attempts at modeling occlusions in *HOG*-style detectors is Wang et al. (2009), leveraging on the separability of block-level responses for a linear *SVM* classifier. A linear *SVM* classifier is represented by a weight vector (learnt on labeled positive and negative training examples), the sign of whose dot product with a test vector indicates the class (positive or negative) of the test vector. It is rather straight-forward to separate the local inner products corresponding to the responses of the blocks in the full window *HOG*. The approach treats the numerical responses of individual blocks as forming an image which is then segmented into sub-regions based on similarity of response. This often results in distinct segments of negative and positive scoring blocks, with the negative scoring segments usually belonging to the occluded portion of the object. If however all the segmented regions comprise of negative scores, then the window is labeled as not having an object. It should be noted that even though *HOG* is a global, rigid model, this approach effectively treats the blocks in the *HOG*-window as rigidly assembled parts. This should not be surprising, since reasoning about partial occlusions is intuitively better suited with part-based models.

Poselets. One important approach called *poselets* (Bourdev and Malik, 2009) replaces local invariant features used in Hough-style detectors as the visual words, with larger rigid *HOG* templates. For inference it obtains consensus over part evidence as in ISM. A large number of training images are labeled at the level of local parts (*e.g.* elbow joint, neck, head, hand). The training set is then automatically divided into many clusters, each consisting of the same set of parts in similar 2D layout, so that *e.g.* all upper body examples in the “waving” pose get assigned to a single cluster. Next, separate rigid HOG-like templates (*poselets*) are trained for each of these clusters. At run time, the entire bank of poselet detectors is evaluated on the test image and high-scoring activations are agglomerated together to give the final detections. A major strength of *poselets* lies in the fact that it can even deal with severely articulated objects like humans in a wide variety of poses.

We utilize our own implementation of *poselets* (but using *DPMs* instead of rigid templates as base detectors) to obtain coarse 2D bounding boxes as well as additional evidence for our local parts locations. The second layer of our model is a detailed 3D wireframe model equipped with an explicit occluder representation based on a set of pre-defined occluder

masks (Chapter 4). The set of occluder masks together with a neighborhood function, act as a prior distribution representing plausible occlusion patterns. An occlusion mask defines which parts are visible, and which ones are not. For the invisible parts behind the mask, image evidence is then not considered. Since the occluder is represented explicitly the framework is not limited to occlusions inferred from a lack of evidence, but also covers occlusions by other detected objects (Chapter 5).

1.3.4 Detailed 3D object modeling

Most current object models output coarse estimates like 2D or 3D bounding boxes around objects in images. Such object hypotheses convey very little geometric information and bounding boxes always over-estimate object extent, which makes high-resolution reasoning about object interactions difficult. However, historically a number of rich 3D object models were proposed, starting with the work of Roberts (1963) which considered images of synthetic polyhedral objects without any real-world clutter in background or foreground. The approach reasons in a 3D reference frame, considering self-occlusions, as well as compositions of complex objects from simpler shapes. However line based object models are not representative of most real-world object classes, and having no mechanism to reject the many spurious edges that are inevitably present in any uncontrolled environment means that the system does not work outside controlled settings. Another classical system is the *SCERPO vision system* of Lowe (1987) which matches a 3D CAD model of a specific polyhedral object to a cluttered image containing multiple instances of the same object, which mutually occlude each other. The method automatically separates line segments in the (edge) image, and utilizes line grouping cues (collinearity, closeness, and parallelism) to form a large set of initial object hypotheses. The groupings are ranked according to pre-defined rules, and finally edge-based matching against projection of the 3D CAD model is performed to refine the hypotheses and obtain a fine-grained hypothesis about the scene. Again while the system performs well for a few test images, it is only applicable to controlled settings, because realistic images have far too many distractors in the background to robustly reason at the level of short line segments.

A very recent work that has similar characteristics to our detailed 3D object model, is the *Aspect Layout Model (ALM)* of Xiang and Savarese (2012). This is a part-based model with planar segments representing the parts (*e.g.* 1 planar segment in the bicycle class, 6 in the car class), which are defined manually for each object class. Relative part locations in 3D are learned on manually annotated CAD models. The appearances of the parts are learnt as *HOG* templates over renderings of 3D CAD models. Like many other approaches including ours, they organize detection as a two stage process, by training so called *root* templates trained over full object views as a coarse first stage, iteratively refined by deforming the part templates. Since the parts are planar, viewpoint invariance is achieved by rectifying each planar segment to its frontal pose. They demonstrate competitive results for viewpoint and 2D part localization accuracy, over an impressive array of 16 different object classes. On the other hand, their model is coarser than our deformable wireframe model (21 parts in the bicycle class, 36 in the car class) and reasons about object viewpoint in discrete steps as compared to our continuous viewpoint reasoning, which impede high-resolution reasoning about object interactions.

They further extend this model to perform scene-level reasoning in Xiang and Savarese (2013), discussed in Sect 1.3.6.

Another very recent example of fine-grained object modeling is the approach of Hejrati and Ramanan (2012), which also uses a two stage approach. However here all fine-grained reasoning is performed in 2D (first stage). This is achieved by extending the DPM model, with a global mixture model to enforce globally plausible geometry and allow self-occlusion reasoning across viewpoints. In the second stage, the 2D hypotheses are lifted to 3D by using a standard non-rigid Structure from Motion (SfM) algorithm (Torresani et al., 2003), assuming correspondences between the 2D part location estimates and a fixed 3D wireframe. Even though the 3D lifting seems like an after-thought in this model, it enforces stronger global geometry constraints, causing part-level localization accuracies specially for occluded parts to improve. All learning is performed on 2D images, as opposed to 3D CAD data often used for such modeling. The advantage of staying close to the DPM model is that the inference is based on dynamic programming and thus much faster than both ALM and our model. Another obvious advantage over the ALM and our initial model (Chapter 3) is the implicit occlusion reasoning. However the approach utilizes a rather rigid 3D model and thus does not provide accurate object-level reconstructions, as we do.

Our detailed wireframe model is closer to Hejrati and Ramanan (2012) w.r.t. part representation (point parts which are vertices of a 3D wireframe), but more similar to the ALM (Xiang and Savarese, 2012) w.r.t. training data (3D CAD models). Our inference procedure is also closer to the latest "scene-level" version of ALM (Xiang and Savarese, 2013), since both are based on stochastic simulation. While a dynamic programming based inference as in Hejrati and Ramanan (2012), reaching the global optimum in much lesser time is obviously preferable, we are unaware of any inference procedure that registers detailed 3D models with 2D image data without some form of "hypothesize-and-verify" step which is always expensive. In fact, fine-grained 3D modeling has been explored in more detail in the face recognition community, and still their inference algorithms remain very slow ($\approx$20 minutes to fit one face model in the recent work of Schönborn et al. (2013) which utilizes a similar detailed representation to ours). On the contrary, our object model is more fine-grained as compared to both of these state-of-the-art models, allowing object-object interactions to be modeled at the level of small parts like a bumper corner on a car or a saddle on a bicycle, and reasoning in a continuous 3D space.

1.3.5 Coarse scene modeling and context

Starting with Hoiem et al. (2005), a number of interesting approaches have been revived which reason about scene layout in 3D. With this research, a broad concensus has emerged that coarse modeling in 3D and contextualizing the detection of different scene elements improves all estimates (see Hoiem and Savarese (2011) for an excellent survey and discussion). Here we discuss two such interesting systems: Hedau et al. (2009) which deals with indoor scene interpretation given a single view, and Ess et al. (2009) which merges multiple visual recognition components for understanding outdoor video sequences.

Hedau et al. (2009) interprets an image of a cluttered room, simultaneously modeling the room as a 3D box and segmenting out the clutter in the room. To estimate the orientation of the room box, the approach utilizes the concept of *vanishing points*. A *vanishing point* (Hartley and Zisserman, 2004) is a 2D point in a perspective projection, where the images of parallel 3D lines intersect (due to a perspective camera). They calculate the intersection points for all line pairs extracted from the test image, and use a voting-based filtering approach to minimize angular deviation between the individual line segments and candidate intersection points. This procedure outputs a set of 3 vanishing points, corresponding to projections of 3 mutually orthogonal directions, thus fixing the room orientation. Next, it hypothesizes many box layouts from these points to obtain translation of the (room) box. These hypotheses are ranked by a discriminative regression approach whose parameters are learned (offline) on a set of labeled training images. In an alternating fashion, the algorithm also keeps labeling the clutter (by classifying groups of similar pixels), defined as the objects lying on the floor and the objects occluding the current box outlines. The key insight is that a good estimate of the room box leads to better localization of clutter, and a good localization of clutter in turn leads to better estimates of the room box. They demonstrate clearly superior performance using this joint estimation approach as compared to indepedent estimation - strongly advocating 3D contextual reasoning.

Automotive navigation applications have inspired a number of coarse scene understanding attempts (Geiger et al., 2011; Wojek et al., 2013) in recent years, particularly since the work of Ess et al. (2009). Ess et al. (2009) utilize a moving stereo rig to model a dynamic street scene, integrating a number of modules including object class detectors, ground plane and visual odometry estimation, 3D object tracking as well as occlusion reasoning. Object class detection for cars and bicycles, is achieved by *HOG* detectors - a single detector for pedestrians, and a bank of seven viewpoint-specific detectors for cars. However the approach verifies the 2D detections thus obtained, by enforcing that multiple objects cannot occupy the same 3D space. It models a temporally varying ground plane, whose joint estimation together with objects, improves the accuracy of both these components. For visual odometry, it masks out the detected objects, to use features only from the static portions of the scene thus improving localization accuracy. The approach also models object-object occlusions and leverages this knowledge to keep object hypotheses alive even when they get fully occluded. To this end, it benefits from a motion-model for updating its estimates of even completely hidden objects. Thus the system benefits from the synergy among various visual recognition and estimation modules – empirically demonstrating improved performance for all the individual estimation tasks.

1.3.6 Fine-grained scene modeling

The approaches described in the previous section already give improvements over independent object-level detections, by coarsely modeling in 3D and employing contextual reasoning. However some researchers in the early days of computer vision had already considered reasoning at a much finer granularity than 2D or 3D bounding box representations. Unfortunately these attempts did not provide significant benefit over independent modeling of scene elements, owing to the limited computational capacity of the day which becomes a serious bottleneck as the degrees of freedom of the

configuration space grow, as well as lack of powerful discriminative methods, robust descriptors, and optimization approaches.

One impressive and very detailed system is presented in Haag and Nagel (1999), for the task of traffic scene modeling from an elevated camera. The system employs a calibrated camera together with detailed polyhedral models of vehicles, buildings, traffic poles, and trees. It even includes an illumination model to predict shadows of moving objects as well as static scene elements. The motion of the moving objects is tracked through *optical flow* and *Extended Kalman Filter (EKF)* updates. *Optical flow* represents the 2D motion in image plane that each pixel undergoes between consecutive frames of a video, and an EKF models the motion of an object as a Gaussian distribution, predicting object displacement for the next frame and updating its belief based on image evidence (here, image edge information). Occlusions, shadows, and metric depths are all computed by ray casting in an explicit geometric model of the scene. Except for the tracking process, they needed to provide all the information manually: the initial object detections and correct 3D object model estimates, the scene model (polyhedral models for building, trees, polls), as well the sun light direction for illumination modeling. Unfortunately, matching image edges with model edges is very error-prone and often causes incorrect matches due to spurious background edges, thus even the tracking is not robust. While practically the system does not perform well, clearly the extent of the modeling was far ahead of its time and inspirational.

During the time frame of this thesis, some approaches have been developed, which focus on detailed 3D scene-level modeling and have been successful in leveraging on the greater expressiveness of their models. One notable system is the *Spatial Layout Model (SLM)* (Xiang and Savarese, 2013), which builds upon *ALMs* (Xiang and Savarese, 2012) discussed in Section 1.3.4. Since the large planar segments of ALM are not flexible enough for occlusion reasoning, they are decomposed into smaller “Atomic Aspect Parts (AAPs)”. Like our method (Chapter 5), *SLM* also utilizes a ground plane assumption, albeit in the form of a soft penalty term in the objective function that pulls objects to stand on a common ground plane, as well as a term penalizing overlap between two 3D objects. They additionally bias their objective function towards closer objects (in terms of distance from the camera), and consider 2D object detection scores in deciding the depth ordering (higher detection score likely corresponds to a closer object). Also similar to our first stage, SLM generates many “aspectlets” which are configurations of adjacent AAPs (HOG templates). The aspectlets perform Hough voting to obtain the full object 2D bounding boxes. The only mode of occlusion considered in the system is that by other detected objects (equivalent to our “deterministic occlusion reasoning”, Chapter 5), but no way of modeling partial occlusions caused by unmodeled scene elements (as done by our “searched occluder”, Chapter 4). The inference is based on reversible jump Markov Chain Monte Carlo (RJMCMC) algorithm, which is a key theoretical strength of this approach and allows adding and deleting new objects from the 2D detection stage during inference. On the other hand it may be difficult to reproduce and tune.

Another system that has grown in parallel to this thesis is the “Bayesian room” understanding system of Del Pero et al. (2012, 2013). This system builds on top of a coarse

parallelepiped room model (Hedau et al., 2009) as discussed in Section 1.3.5, adding detailed 3D furniture models inside the room. The objects (furniture) like chairs, beds, tables, cupboards are all represented by configurations of deformable cuboids with appropriate aspect ratios, learnt over training images. They utilize a combination of low-level and high-level cues to model appearances: *(i)* explicit reasoning about edges (presence, absence and noise edges), *(ii)* surface orientations determined by checking alignment against global cuboid (room) orientation, *(iii)* semantic pixel labeling into object, floor, ceiling, walls like Hedau et al. (2009), and *(iv)* self-similarity based on color distribution. Inference is tuned separately for different scene elements *e.g.* to detect peg structures where furniture legs meet the floor. The system further employs high-level contextual relationships among full objects, *e.g.* random sample generation (for inference) is biased to propose chairs near tables. Like Xiang and Savarese (2013), they also use a variation of RJMCMC for inference, allowing addition, deletion, and switching of object hypotheses at inference time, avoiding early commitment to number and type of objects.

Like the present thesis, these very recent approaches highlight the benefits that detailed 3D scene-level reasoning can now bring to visual recognition. Not only do such rich models deliver greater information in terms of 3D scene layout, but joint reasoning is also found to improve individual recognition rates for all the scene elements.

1.4 Relevance to science and economy

The contributions made in this thesis are directly applicable towards the solution of important challenges in robotic perception and planning as well as in augmented reality. In the following we mention some problem domains where our ideas can have an immediate impact.

1.4.1 Markerless Augmented Reality

Most current augmented reality (AR) systems establish alignment between the real and virtual worlds through well-textured planar templates (Chen et al., 2009; Wagner et al., 2010). However the world is three dimensional and most object classes are not well-textured. The applicability of AR systems can be widely extended if fine-grained 3D geometric structure of object classes and scenes from images can be recovered. Recent work of Hengel et al. (2007) attempts to extract such detailed higher-level 3D models for specific object instances and scenes but it requires video (to reconstruct the scene) as well as user interaction. The techniques developed in this thesis allow semantic recognition, 3D pose estimation, and recovery of plausible 3D geometry even for partly occluded objects from single images, apart from scene-level reasoning and provision of supporting-plane hypotheses. Such detailed interpretation enables accurate registration of virtual information into complex real-world images. Since the system does not require complex sensing modalities like laser range sensing, and is massively parallelizable in its current form, it has the potential to be useful for mobile AR applications.

Economic potential. Mobile AR is seeing rapid development and turning into a big business, with heavy investment coming from industry giants such as Qualcomm. As opposed

to approaches based on markers or explicit scene reconstruction which can only be applied under controlled settings, semantic reconstruction can be applied anywhere *e.g.* to project animated characters into a street scene in real-time, or provide live high-resolution interactive experiences in a classroom environment. Thus there exists potential to leverage on this relatively unexplored technique (*i.e.* semantic reconstruction) in the context of augmented reality applications.

1.4.2 Mobile robotics - localization and mapping

Simultaneous Localization and Mapping (SLAM) is a technique to build a map of an unknown environment as a mobile robot explores it, while at the same time localizing the robot w.r.t. the map. Most SLAM research is focused on mapping and tracking low-level features, such as interest points (Davison et al., 2007; Klein and Murray, 2007), or reconstructing denser representations[2] comprising of point clouds or geometric meshes as in Newcombe et al. (2011). After two decades of exclusive focus on "low-level" SLAM approaches, interest is now shifting towards integration of semantic concepts into SLAM pipelines by incorporating planar structures (Schindler and Bauer, 2003; Gee et al., 2008), coarse 2D object class detections (Cornelis et al., 2008; Bao and Savarese, 2011), and even specific 3D object detections (Fioraio and Stefano, 2013; Salas-Moreno et al., 2013). This interest springs from the fact that it is still not possible to accurately reconstruct textureless objects such as walls, and highly specular objects such as cars, and problems such a multi-body SLAM (SLAM for dynamic scenes) can benefit a lot from higher-level regularization. Besides, use of higher-level objects as building blocks causes the representation to get much sparser and localization accuracies tend to improve. Thus, the detailed 3D object class model developed in this thesis, as well as higher level reasoning (*e.g.* deterministic occluder reasoning) can eventually be integrated into SLAM pipelines.

Economic potential. There are huge investments being made in the domain of autonomous driving and driver assistance systems by all big automotive manufacturers as well as technology giants like Google. Unfortunately, the most successful practical examples remain the ones utilizing pre-mapped environments and prohibitively expensive laser range sensing (*e.g.* Google's initiative which is based on the Velodyne sensor). To make the technology economically feasible and broadly applicable (*e.g.* in regions where such dense maps may not be available), there is a need to strengthen strengthen 3D scene reasoning from monocular or stereo cameras. Thus our semantic 3D scene understanding approach is particularly relevant, even more so since our experimentation has focused on street scenes.

1.4.3 Metrology and content-based search

Further practical applications that may be facilitated include image-based metrology and content-based retrieval. Consider *e.g.* forensics, where in the absense of planned photogrammetric record, 3D reconstruction from single random photographs, could be

[2] also called Structure from Motion or SfM

helpful in detailed understanding of a scene.

In the context of digital 3D databases, such models could help in a myriad of ways for searching objects in 3D databases; possibly even enhancing such databases by allowing learning model characteristics such as texture and partial shape from images (Zia et al., 2009).

1.4.4 Scientific and Industrial recognition

The work presented in this thesis has received several awards and scholarships from both academia and industry: (i) a *Best Paper Award* from Microsoft Research for Zia et al. (2011), (ii) a *Best PhD Student award* for an extended poster on Zia et al. (2013) from International Association for Pattern Recognition (IAPR), and (iii) a *Qualcomm Innovation Fellowship 2012* (worth 10,000 EUR) for the initial proposal of Zia et al. (2014a,b).

Chapter 2

Background

We have already explored the key ideas from literature that are relevant for this thesis. In this chapter, we introduce the tools that we build upon in the upcoming "core" chapters (*i.e.* Chapter 3, 4, 5). While complete text books are available discussing each of these tools in great depths, we tailor our treatment to the aspects most relevant to understanding the system developed in this thesis. The discussion starts with local image descriptors which are used in our approach to represent the local appearance of the parts of our object classes, such as the appearance of the wheels of a car or the handle of a bicycle. We further discuss discriminative classification techniques, which are later used to actually detect these parts in a test image. Next, we describe the detailed 3D geometry model used to represent object classes in our approach. And finally, we specify the basic inference algorithm that we build upon in later chapters to detect and reconstruct full object instances and reason about their interactions.

2.1 Local image descriptors

Local image patches, represented by image descriptors, form the basis of an important portion of computer vision applications: image matching (Tuytelaars and Gool, 2000; Brown and Lowe, 2003), object detection (Leibe et al., 2006; Liebelt et al., 2008), and texture recognition (Lazebnik et al., 2003), to mention just a few. An image patch can be "described" directly by a vector of pixel intensities, as sometimes done to establish correspondences between points in images of the same scene, by computing sum of absolute differences, sum of squared differences, or normalized cross-correlation between these vectors. However, it is often preferable to encode the patch by an attribute that is relevant to the task at hand. For example in the case of texture classification, the spatial frequency content of the image patch is important, whereas for object detection, describing the local shape which usually comprises of a non-repetitive structure, by gradient orientations is more informative. Furthermore, descriptors should be invariant to relevant imaging variations, such as lighting changes or local affine transformations. An array of descriptors tuned for different tasks have been proposed in the literature (see Mikolajczyk and Schmid (2005) for a detailed survey). We already mentioned the *Histogram of Oriented Gradients (HOG)* descriptor of Dalal and Triggs (2005), which is a global descriptor for a larger window, encoding object class appearance in Section 1.3.1. The most well-known local descriptor is *Scale Invariant Feature Transform (SIFT)* of Lowe (2004), which en-

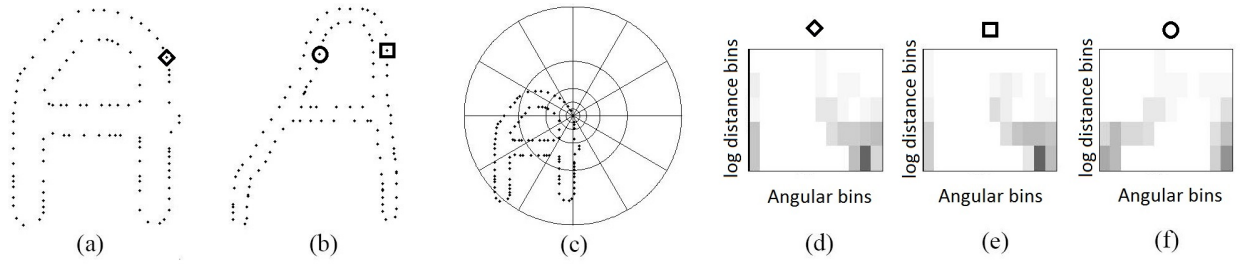


Figure 2.1: Shape Context descriptor. (a) and (b) show two example shapes. (c) visualizes the histogramming for location marked ♢ in (a). (d), (e), and (f) visualize the histograms evaluated at locations marked by ♢, □, ◯ in (a) and (b). Illustration reproduced from Shape Context (2013) published under Creative Commons license.

codes an image region by histograms of gradient orientations computed over rectangular sub-regions inside the image patch of interest. For each sub-patch, a separate gradient orientation histogram is computed, which are combined together to form the descriptor for the image patch. The vector is normalized to a unit vector to achieve some degree of invariance to illumination changes. Here we discuss the *Shape Context (SC)*, introduced by Belongie et al. (2000). We use a variant of SC from Mikolajczyk and Schmid (2005) to represent local part appearances in our system (Chapter 3).

2.1.1 Shape Context

The original *Shape Context* descriptor (Belongie et al., 2000) at an edge pixel (point of interest), is a histogram of relative coordinates of randomly sampled points on the shape. The relative coordinates of the sampled points are transformed to log-polar space, to fill bins that are defined according to a log scale. The logarithmic distance binning is motivated by the increase in positional uncertainty as the distance from the point of interest increases, *i.e.* we expect nearer points on the shape to maintain their relative position, however farther away points are allowed to be displaced more. This allows the representation to be robust to locally affine transformations of the shape. The approach uses 5 bins for logarithm of distance and 12 bins for angular dimension in their original implementation. Figure 2.1 shows a toy example illustrating the approach.

We use the implementation of Mikolajczyk and Schmid (2005) to represent the local appearance of our object parts (Chapter 3). This variant captures the distribution of local gradient orientations as opposed to counting the relative edge pixel locations, on the same log-polar histogram. The pixel intensities for the patch are normalized by adjusting their mean and variance to fixed values to make the descriptor invariant to illumination changes. The performance of part-level localization in the context of fine-grained object class recognition is compared for different descriptors (including different variants of SIFT) in Andriluka et al. (2011), in a setting very similar to ours. They show superior results for this variant of shape context descriptor, which they attribute to finer gradient discretization, use of gradients as opposed to edge-based features, and log-polar spatial binning.

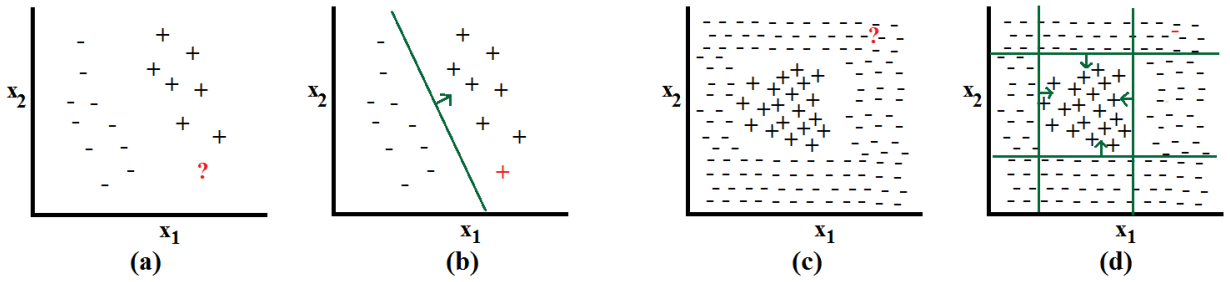


Figure 2.2: Classification. (a) Linearly separable classes, (b) linear classifier, (c) more realistic example without linearly separable classes, (d) combination of linear classifiers to perform non-linear classification.

2.2 Classification

A *classifier* is a function $f(\mathbf{x}) : \mathcal{R}^d$ x $\mathcal{C}$, which predicts the class label $y = f(x)$ of an input data point $\mathbf{x} \in \mathcal{R}^d$, out of a set of discrete class labels $\mathcal{C} = \{C_1, C_2, ..., C_M\}$, after learning the intended class memberships from a set of labeled training examples $\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_N}, y_N)\}$. One example application is to predict whether or not it will rain on a certain day based on temperature, humidity and pressure; learnt on many past observations.

The simplest case is that of linearly separable classes, which can be separated by a hyper-plane in $\mathcal{R}^d$. Figure 2.2 (a) and (b) visualizes such a case, for $d = 2, M = 2$. However, the more general case is that of classes which are not linearly separable (Figure 2.2 (c)). One way of handling such cases is to approximate the non-linear classification boundaries as a combination of simple (*e.g.* linear) classifiers (Figure 2.2 (d)). We describe two such algorithms below, which we utilize in Chapter 3 to classify local image patches (encoded as dense shape context descriptors), as one of the object's parts or background.

2.2.1 AdaBoost

AdaBoost is an algorithm introduced by Freund and Shapire (1996), which trains a number of "weak classifiers" (such as axis-aligned one-dimensional decision thresholds) on the training examples. The classifiers are trained sequentially by iteratively adjusting the weights of data points based on their classification accuracy with the previous classifier.

We describe the procedure as Algorithm 1, following the notation of Bishop (2007). The basic algorithm considers two classes labeled from $\mathcal{C} = \{-1, +1\}$. The weight of the training data points $\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_N}, y_N)\}$ are specified by w_n. The key idea is to train weak classifiers $f^{(t)}(\mathbf{x})$ one after another, and increase the weights of misclassified data points before the next iteration, to focus the next weak classifier on the "difficult" data points..

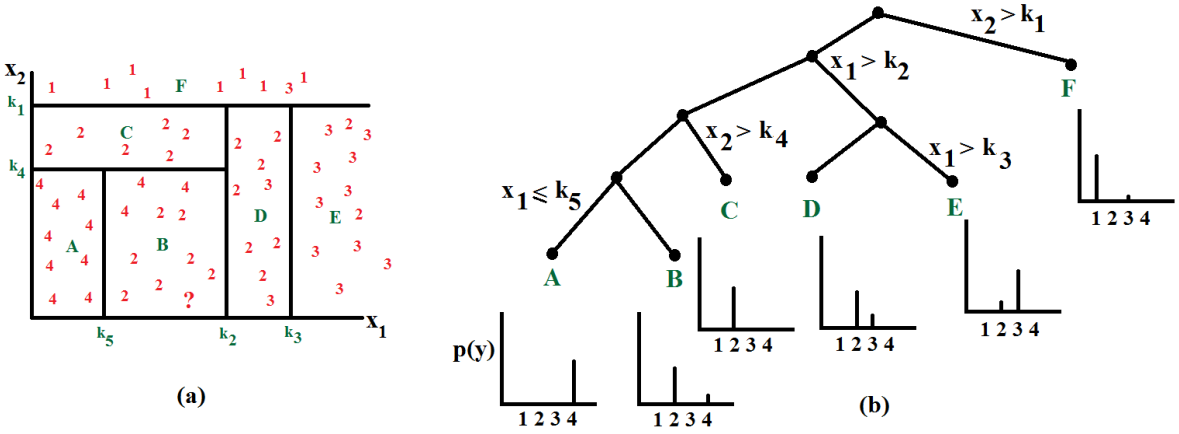


Figure 2.3: Decision tree classifier. (a) A 2-dimensional input space with the training examples for a 4-class case, and recursive 1-dimensional decision stumps visualized. (b) Decision tree learned over the training data visualized, where each node represents a portion of the input space. The non-leaf nodes also have an associated decision stump. The likelihoods of class-membership for each leaf node are also visualized.

At test time, the class of an unseen data point is the weighted sum of predictions from weak classifiers, using classifier weights $\alpha^{(t)}$. Specifically, the class of a test data point $\mathbf{x}$ is described by,

$$y = f(\mathbf{x}) = \text{sign}\Big(\sum_{t=1}^{T} \alpha^{(t)} f^{(t)}(\mathbf{x})\Big) \tag{2.1}$$

The algorithm is very straight-forward to implement and requires almost no parameter tuning (except maximum number of iterations T). However the approach cannot directly handle multiple classes, as required in the context of classifying an image patch into one of many object parts. Extensions to achieve multi-class classification exist, but they are slow and not principled. Further, we empirically found that it cannot model part appearances as seen from different viewpoints as a single class (Chapter 3; Zia et al., 2011). In our investigations, we found another technique to be more suitable for our problem, namely the *Random Forest* classifier which we discuss next.

2.2.2 Decision trees and Random Forest

Another classification scheme that can combine weak classifiers to approximate highly non-linear and multi-modal class boundaries is the *decision tree* architecture (Breiman, 1984; Quinlan, 1986). A decision tree represents a recursive binary partitioning of the input space, as illustrated in Figure 2.3. It uses a simple decision (such as a one-dimensional decision stump) at each non-leaf node of the tree. Classification is performed by "dropping" down the test data point from the root, and letting it traverse a path decided by the node decisions, until it reaches a leaf node. Each leaf node has a corresponding probability distribution (learnt on training data), which specifies the likelihood of the test point belonging to different classes. In Figure 2.3 (a) the class memberships for the labeled training examples are shown, on which the tree structure and class-membership likelihoods in (b) are learned. For the unseen test point, represented by the question

Input: Training set: $\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_N}, y_N)\}$ with $y_n \in \mathcal{C} = \{-1, +1\}$.
Output: Set of T simple classifiers $\{f^{(1)}(\mathbf{x}), f^{(2)}(\mathbf{x}), ..., f^{(T)}(\mathbf{x})\}$ and corresponding weights $\{\alpha^{(1)}, \alpha^{(2)}, ..., \alpha^{(T)}\}$.

1. Initialize weights of training data points, $w_n^{(1)} = \frac{1}{N}$

2. **for** *fixed number of iterations:* $t = 1, ..., T$ **do**

(i) Learn a simple classifier $f^{(t)}(\mathbf{x})$ on the training set, maximizing the sum of weights corresponding to correctly classified data points:

$$\sum_{n=1}^{N} w_n^{(t)} I\big(f^{(t)}(\mathbf{x}_n) \neq y_n\big)$$

where $I\big(f^{(t)}(\mathbf{x}_n) \neq y_n\big)$ is equal to 1 for an incorrectly classified data point $f^{(t)}(\mathbf{x}_n) \neq y_n$ and 0 for $f^{(t)}(\mathbf{x}_n) = y_n$.

(ii) Calculate the normalized weight $\alpha^{(t)}$ for the simple classifier learned in the current iteration:

$$\epsilon^{(t)} = \frac{\sum_{n=1}^{N} w_n^{(t)} I\big(f^{(t)}(\mathbf{x}_n) \neq y_n\big)}{\sum_{n=1}^{N} w_n^{(t)}},$$

$$\alpha^{(t)} = ln\left\{\frac{1 - \epsilon^{(t)}}{\epsilon^{(t)}}\right\}$$

(iii) Re-calculate the weights of training data points, based on current classifier weight $\alpha^{(t)}$ as well as whether or not a training example is correctly classified with the current classifier:

$$w_n^{(t+1)} = w_n^{(t)} \text{exp}\Big\{\alpha^{(t)} I\big(f^{(t)}(\mathbf{x}_n) \neq y_n\big)\Big\}$$

end

Algorithm 1: AdaBoost algorithm training (based on the notation of Bishop (2007)).

mark in Figure 2.3 (a), it is straight-forward to start at the top of the tree in Figure 2.3 (b) traversing the tree by following the binary decisions at the nodes to reach a leaf node *B*. Here we see that the corresponding region in the input space is *B*, for which the most likely class membership is Class 2.

Learning tree structure. Estimating the optimal structure including the number of nodes, the dimension of data point to be considered at each node, as well as the corresponding threshold to minimize training error is computationally infeasible for most problems of practical interest. Thus, a greedy approach is often utilized starting at the root node, and growing the tree by sequentially adding nodes which take locally optimal decisions on the training data (Bishop, 2007). Each new node corresponds to a region of the input space, divided into two sub-regions by the decision function on the node. To estimate the optimal decision function that best divides the region, a local search is performed (often exhaustive) for an input space dimension and the corresponding threshold value which optimizes a measure of class segmentation at the node. The local measures of class segmentation which are most commonly used in this context are *cross-entropy* and the *Gini index*. Following the notation of Bishop (2007), let $p_{\tau C_m}$ represent the proportion of training data points in Region τ belonging to class C_m, where $m = 1, ..., M$, and k be the threshold value, then *cross-entropy* is described as

$$Q_\tau(k) = \sum_{m=1}^{M} p_{\tau C_m} ln\big(p_{\tau C_m}\big) \tag{2.2}$$

and the *Gini index*

$$Q_\tau(k) = \sum_{m=1}^{M} p_{\tau C_m}\big(1 - p_{\tau C_m}\big) \tag{2.3}$$

Different approaches can be utilized to decide when to stop adding further nodes, *e.g.* stopping when the training classification error falls below a threshold or when a certain maximum depth is reached.

Bagging and Random Forests. Breiman (1996) and Amit and Geman (1997) proposed combining de-correlated classifiers, which is called *Bagging* (short for Bootstrap aggregating), and showed that it helps with the problem of overfitting to training data and thus improves classification accuracy on unseen data points. *Decision trees* individually tend to overfit, but they are efficient so running multiple trees is feasible, and it is straight-forward to average their outputs. Such an ensemble of multiple decision trees is called *Random Forest*.In order to learn multiple de-correlated trees (Breiman, 2001) over a training set, randomization can be employed at different points in the tree learning process. This includes learning different decision trees on randomly picked subsets of the training set, or randomly choosing the input space dimension in which to perform the local decision at different nodes. The results can be combined by averaging the class-conditionals from the individual trees. Random Forests are inherently multi-class, readily parallelizable owing to independently grown and evaluated trees, and straight-forward to tune. In our investi-

gations (Chapter 3), we also found them well suited to model multi-modal distributions. In recent years, Random Forests have been increasingly applied to computer vision problems (see Leistner (2010) for a treatment focused on computer vision applications).

2.3 Point-based shape analysis

Reconstructing 3D object shape from a single view is an ill-posed problem, and needs strong shape priors. Unfortunately the part-based models most commonly used in the context of object class detection (Section 1.3.1) only loosely model object shape, *e.g.* as a star topology with the part locations learned relative to a root part as Gaussian distributions (Stark et al., 2010). Thus such models do not constrain the overall object geometry enough to recover it from noisy part detections, and are limited to providing 2D bounding box detections only. One successful technique for detailed shape modeling is the classic *Active Shape Model (ASM)* of Cootes et al. (1995), which strongly constrains the relative part locations, always outputing globally plausible geometry as learnt on a training set. It provides a deformable wireframe representation based on a set of vertices of the object class of interest. The allowable global deformations are described by a handful of displacement vectors for all vertices, which are then linearly combined. We describe this approach in detail next.

ASM learns a shape basis on a set of training shape exemplars for an object class with a well-defined topology. Each aligned shape exemplar is represented by a vector of relative vertex locations, with each vertex representing the location of a well-defined "landmark" point on the shape, *e.g.* corners of bumper, fenders, roof, centers of wheels, and so on for representing the shape of the car class. A weighted sum of the shape basis vectors equals a new shape within the shape space defined by the training shape exemplars.

Let the shape of n training exemplars be represented by the relative 3D locations of their m vertices as: $\mathbf{x_i} = [x^1_{i,x}, x^1_{i,y}, x^1_{i,z}, x^2_{i,x}, x^2_{i,y}, x^2_{i,z}, ..., x^m_{i,x}, x^m_{i,y}, x^m_{i,z}]^T$, where $i \in \{1, 2, ..., n\}$. The exemplars are aligned to each other and centered at the origin.

We can calculate the mean shape $\boldsymbol{\mu}$ as,

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x_i} \tag{2.4}$$

and the mean-shifted shape examples as,

$$\widetilde{\mathbf{x}_\mathbf{i}} = \mathbf{x_i} - \boldsymbol{\mu}. \tag{2.5}$$

We next collect the training shapes in a matrix,

$$\widetilde{\mathbf{X}} = [\widetilde{\mathbf{x_1}}, \widetilde{\mathbf{x_2}}, ..., \widetilde{\mathbf{x_n}}]^T \tag{2.6}$$

and calculate the covariance matrix as,

$$\mathbf{C} = \frac{1}{n-1} \widetilde{\mathbf{X}} \widetilde{\mathbf{X}}^T. \tag{2.7}$$

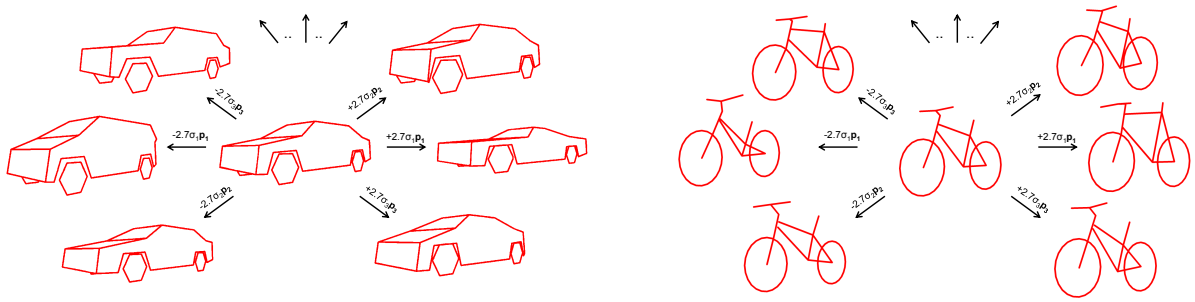


Figure 2.4: 3D Deformable Wireframe models for cars and bicycles. Shapes are sampled from shape model, and the vertex locations connected by manually defined edges.

The eigen-vectors $\mathbf{p_k}$ of the covariance matrix $\mathbf{C}$ represent the principal shape deformations for the object class represented by the training exemplars $\mathbf{x_i}$, which are ranked in order of importance by arranging the corresponding eigen-values. Thus, σ_1 represents the largest standard deviation corresponding to the eigen-vector $\mathbf{p_1}$ which represent the direction of shape deformation along which the largest variation exists in the training dataset; σ_2 and $\mathbf{p_2}$ represent the second largest shape deformation direction, and so on. The eigen-vectors corresponding to small eigen-values usually represent noise variations, and can be discarded. We can represent the Eigen-value decomposition as,

$$\mathbf{C} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \tag{2.8}$$

where, $\mathbf{Q} = [\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_{3m}}]$ and $\mathbf{\Lambda} = diag(\sigma_1^2, \sigma_2^2, ..., \sigma_{3m}^2)$.

Using the principal component directions, and a set of shape weights $s = \{s_1, s_2, ..., s_k\}$ we can then synthesize any shape within the shape space even by using an $r < 3m$ (which causes some variation in the training data set to be neglected),

$$\mathbf{X(s)} \approx \boldsymbol{\mu} + \sum_{k=1}^{r} s_k \sigma_k \mathbf{p}_k \tag{2.9}$$

Setting shape weights $|s_k| < 3$, results in shapes which are within the variations represented in the training set, since 3 times standard deviation from the mean corresponds to 99.7% of the probability mass for a Gaussian. Figure 2.4 visualizes the learned shapes, for cars and bicycles, trained on a set of labeled 3D CAD models. The car and bicycle wireframes in the middle are the mean μ shapes for the respective models.

2.4 Smoothing-based optimization

Whenever a task is defined well enough to formulate it mathematically (as an objective function), finding the “best” solution is called optimization. The standard methods for local optimization include 1st-order approaches (based on Jacobian of the objective), which starting from an initialization take successive steps proportional to the local gradient of the function, and 2nd-order approaches (based on Hessian of the objective) which locally approximate the objective function as a quadratic. Further, many approximations to these

methods such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Broyden et al., 1973; Shanno, 1970) have been proposed which provide various useful properties, such as cheaper computation. However, these approaches cannot reach the global optima for any but a few restricted families of functions, such as quasi-convex functions in the continuous domain (Boyd and Vandenberghe, 2004).Unfortunately, no approach exists which can guarantee reaching a global optimum (within reasonable time) for many types of objective functions of practical value. A number of meta-heuristic search methods have been proposed which are found useful in many cases providing good local optima despite having weak theoretical justifications, such as Tabu search (Glover, 1986) and Evolutionary algorithms (Rechenberg, 1971; Holland, 1975). Besides, some algorithms which possess better theoretical properties (related to convergence and input space coverage) such as the *Monto-Carlo* methods (Metropolis and Ulam, 1949; Metropolis et al., 1953) and the *Particle filter* (Isard and Blake, 1998) have also been proposed for this task, which have been explored more in the context of computer vision problems.

We also encode our requirements for 3D deformable wireframe fitting over image evidence, in the form of objective functions (Chapters 3, 4, 5), which are defined over a mixed set of both continuous and discrete variables, and are high-dimensional and non-convex. Thus, we resort to a simulation-based optimization scheme, where starting from a set of initializations, we iteratively refine the initializations in a stochastic gradient descent procedure, inspired by the *Smoothing-based Optimization* algorithm of Leordeanu and Hebert (2008). We next describe this algorithm, and provide an intuitive motivation without a formal background (which can be found in the original paper).

The algorithm is motivated by two key ideas, which we describe in the following:

1. Smoothing out weak optima. Smoothing the objective function with a Gaussian kernel wipes out shallow local minima due to noise, making it easier to localize the significant ones. Figure 2.5 illustrates the idea on a function with many local optima, for increasing variance of the Gaussian smoothing kernel. Note how a kernel with greater variance causes a greater number of local optima to disappear. Unfortunately, since the Gaussian kernel has infinite support, the entire space needs to be visited even to smooth the original function at a single point. Thus, the idea cannot be applied to optimization in its direct form, and we instead resort to locally sampling the objective function and directly computing a smoothed estimate of a strong local optimum.

2. Iterative update. The algorithm represents its current estimate of a local optimum with a Gaussian distribution. The estimate is refined iteratively, by evaluating the objective function at points in the region of high probability mass of the Gaussian distribution. These evaluations are used to improve the Gaussian estimate and move it closer to a strong optimum value.

Algorithm 2 presents the approach explicitly, optimizing over an objective function $f(x)$, and Figure 2.6 visualizes an example objective function as well as the Gaussian distribution representing the current estimate of a strong local optimum, at different iterations.

Initialization: Mean and variance of Gaussian distribution representing our estimate at iteration $t = 0$: $\mu^{(0)}$ and $\sigma^{(0)}$,.

while $\sigma^{(t)} < \epsilon$ **do**

(i) Draw a set of samples $\{s_1, s_2, ..., s_m\}$, from the normal distribution $\mathcal{N}(\mu^{(t)}, (\sigma^{(t)})^2 I)$.

(ii) Set: $\mu^{(t+1)} = \dfrac{\sum_{k=1}^{m} s_k f(s_k)}{\sum_{k=1}^{m} f(s_k)}$, $\sigma^{(t+1)} = \sqrt{\dfrac{\sum_{k=1}^{m} (s_k - \mu^{(t)})^2 f(s_k)}{\sum_{k=1}^{m} f(s_k)}}$

(iii) $t = t + 1$

end

Algorithm 2: Smoothing-based Optimization algorithm, following Leordeanu and Hebert (2008).

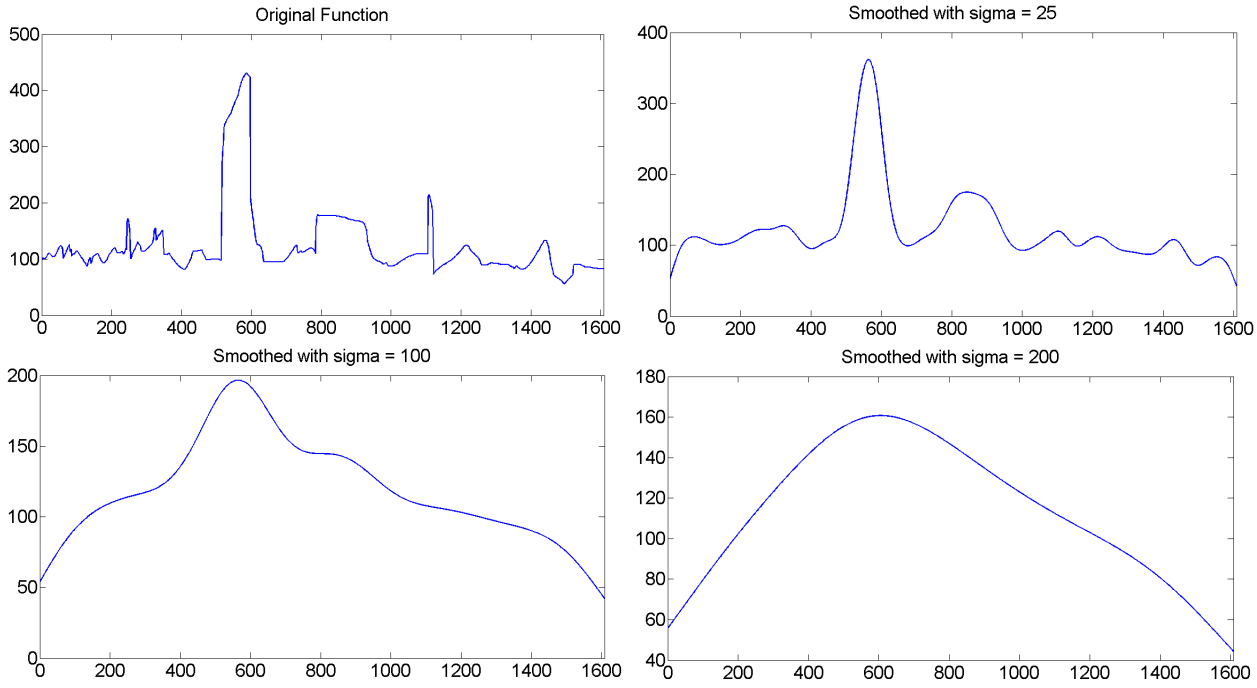


Figure 2.5: Smoothing. As the variance of smoothing Gaussian kernel is increased, more and more of the noisy local optima disappear.

Instead of smoothing over the entire space, we sample over a small region surrounding the current estimate of the optima in line (i). Next, in line (ii), we calculate a mean position $\mu^{(t+1)}$, weighted by the objective function values at the sampled points $f(s_k)$ which displaces our estimate towards a strong peak of the function. The standard deviation $\sigma^{(t+1)}$ of the Gaussian is re-calculated for the estimate, which automatically grows to escape from valleys and narrows when a strong hill is reached. The algorithm terminates when the standard deviation becomes smaller than a threshold ϵ.

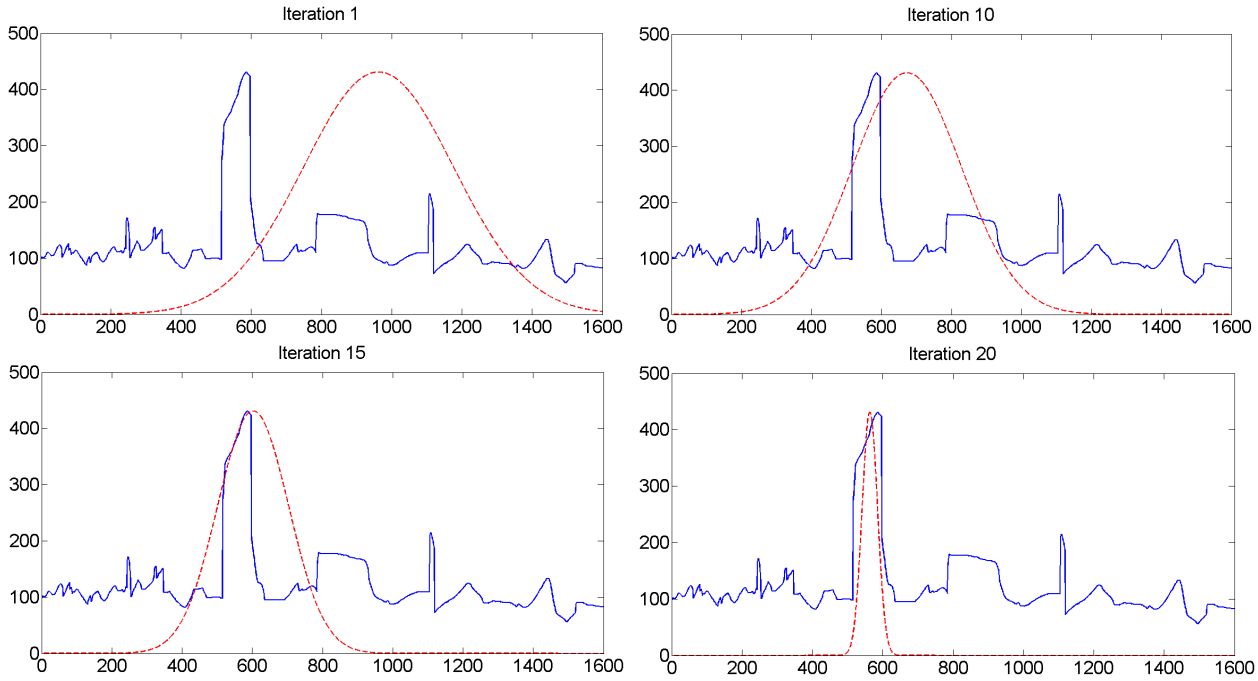


Figure 2.6: Algorithm iterations. Objective function visualized in blue. Estimate of maxima location represented by Gaussian distribution as a dashed red curve.

Chapter 3

Detailed 3D Representations for Object Modeling and Recognition

M. Zeeshan Zia, Michael Stark, Bernt Schiele, Konrad Schindler
IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11):2608-2623, 2013

(Author version; for typeset version please refer to the original journal paper.)

3.1 Abstract

Geometric 3D reasoning at the level of objects has received renewed attention recently, in the context of visual scene understanding. The level of geometric detail, however, is typically limited to qualitative representations or coarse boxes. This is linked to the fact that today's object class detectors are tuned towards robust 2D matching rather than accurate 3D geometry, encouraged by bounding-box based benchmarks such as Pascal VOC. In this paper, we revisit ideas from the early days of computer vision, namely, detailed, 3D geometric object class representations for recognition. These representations can recover geometrically far more accurate object hypotheses than just bounding boxes, including continuous estimates of object pose, and 3D wireframes with relative 3D positions of object parts. In combination with robust techniques for shape description and inference, we outperform state-of-the-art results in monocular 3D pose estimation. In a series of experiments, we analyze our approach in detail, and demonstrate novel applications enabled by such an object class representation, such as fine-grained categorization of cars and bicycles according to their 3D geometry, and ultra-wide baseline matching.

Keywords: 3D Representation, recognition, single image 3D reconstruction, scene understanding, ultra-wide baseline matching

3.2 Introduction

Over the last decade, automatic visual recognition and detection of semantic object classes have made spectacular progress. It is now possible to detect and recognize members of a semantic object categories with reasonable accuracy. Based on this development, there has been a renewed interest in high-level vision and scene understanding, *e.g.* Hoiem et al. (2008); Ess et al. (2009); Wang et al. (2010); Hedau et al. (2010); Gupta et al. (2010); Barinova et al. (2010); Wojek et al. (2010).

The present work starts from the observation that although modern object detectors are very successful at finding things, the object hypotheses they output are in fact extremely crude: typically, they deliver a bounding box around the object in either 2D image space (Viola and Jones, 2001; Dalal and Triggs, 2005; Felzenszwalb et al., 2010) or 3D object space (Liebelt and Schmid, 2010; Hedau et al., 2010; Payet and Todorovic, 2011). That is, the detected object is represented by a box, which differs from other objects only by its size and aspect ratio. We believe that such simplistic object representations severely hamper subsequent higher-level reasoning about objects and their relations, since they convey very little information about the objects' geometry.

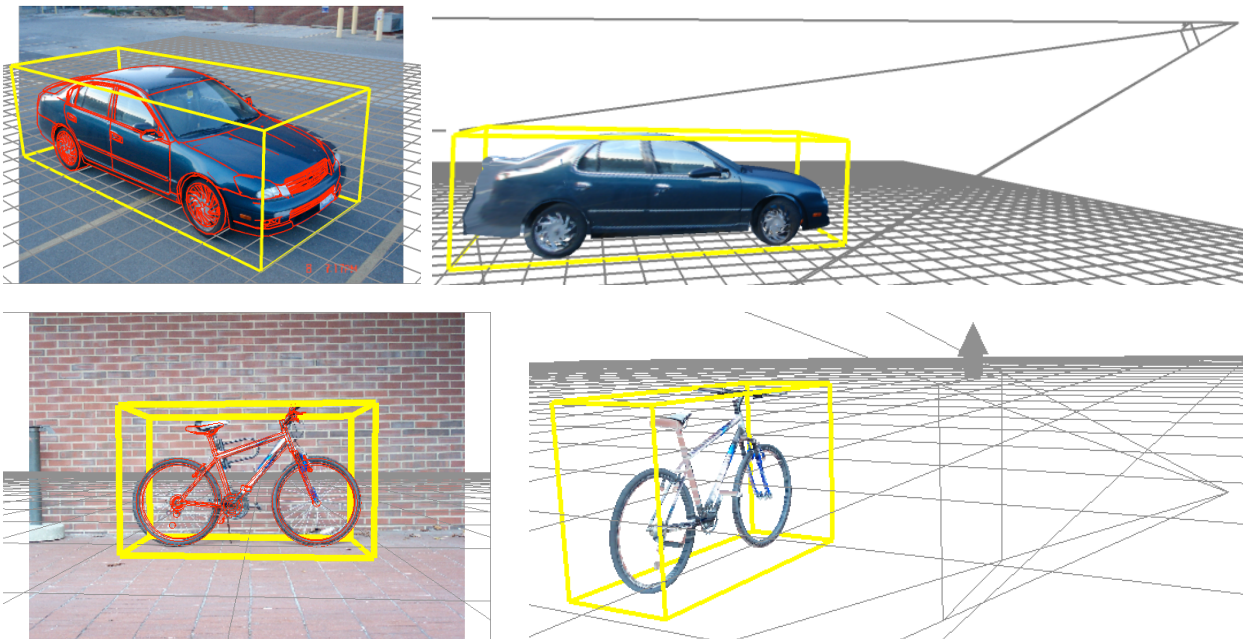


Figure 3.1: Fully automatic shape and pose estimation results. (Left) overlaid closest training 3D CAD model. (Right) reconstruction of object shape, pose, and camera pose (CAD model rendered from novel viewpoint using original image as texture).

We thus try to take a further step towards the ultimate goal of scene-level image understanding, by looking back at ideas from the early days of computer vision. Starting from Marr's seminal ideas (Marr and Nishihara, 1978), many 3D models of objects were proposed, which provided rich and detailed descriptions of object shape and pose (Brooks, 1981; Pentland, 1986; Lowe, 1987; Koller et al., 1993; Sullivan et al., 1995; Haag and Nagel, 1999). Unfortunately, these models proved difficult to match to

real world images. As a consequence, later researchers traded off model accuracy for robustness in matching, for example by representing objects by the statistics of local features in an image window. This has led to impressive performance for recognition of a variety of object classes (Everingham et al., 2010) as well as related tasks like scene classification (Lazebnik et al., 2006), but the extent to which relations between scene entities can be modeled with such representations is rather limited. Also, we note that the recognition performance of 2D appearance representations at present is showing only small improvements and seems to be saturating (*e.g.* at ≈35% average precision for the well-known PASCAL VOC challenge Everingham et al., 2010). Although *per se* this does not mean that more complex models are the way to go, it does raise the question whether some of the difficulties could be overcome with 3D models, which allow one to segment, reconstruct, and recognize in a more integrated fashion.

Over the last couple of years researchers have explored coarse "box-level" representations of 3D geometry in the context of scene understanding (Hoiem et al., 2008; Ess et al., 2009; Wang et al., 2010; Hedau et al., 2010; Gupta et al., 2010; Barinova et al., 2010; Wojek et al., 2010), and have shown that 3D geometric reasoning is not only interesting as a goal in itself, but that the additional information it supplies also leads to better recognition performance. In this work, we try to go one step further. Inspired both by early work on 3D recognition and by more recent advances in 2D appearance descriptors, we combine detailed models of 3D geometry with modern discriminative appearance models into a richer and more fine-grained object representation.

Using a 3D model naturally affords invariance to viewpoint. While viewpoint-invariant detection has been a hot topic for some time now (Schneiderman and Kanade, 2000; Thomas et al., 2006; Yan et al., 2007; Ozuysal et al., 2009; Arie-Nachimson and Basri, 2009; Zhu et al., 2010; Stark et al., 2010; Gu and Ren, 2010; Payet and Todorovic, 2011; Glasner et al., 2011; Villamizar et al., 2011; Pepik et al., 2012b), most approaches are made up of several flat viewpoint-dependent representations connected together in one way or the other. There are some more recent works which model the 3D geometry more explicitly (Bourdev and Malik, 2009; Liebelt and Schmid, 2010; Sun et al., 2010; Chen et al., 2010; Pepik et al., 2012b). While these are an important step towards true 3D recognition, they typically still deliver 2D or 3D bounding boxes as output, and there is still room for improvement in the granularity of the output hypotheses.

System overview. We exploit the fact that for many important classes there are already high-quality 3D models available, and start from a database of 3D computer aided design (CAD) models of the desired object class as training data. After simplifying the raw CAD models we apply principal components analysis to obtain a coarse 3-dimensional wireframe model which captures the geometric intra-class variability. In order to capture appearance, we train detectors for the vertices of the wireframe, which we call "parts". The training is also based on renderings of the (original, unsimplified) CAD models, such that our model does not require any image annotation. We apply the model to two rather different object classes, cars and bicycles.

At test time, we generate evidence for the parts by densely applying the part detectors

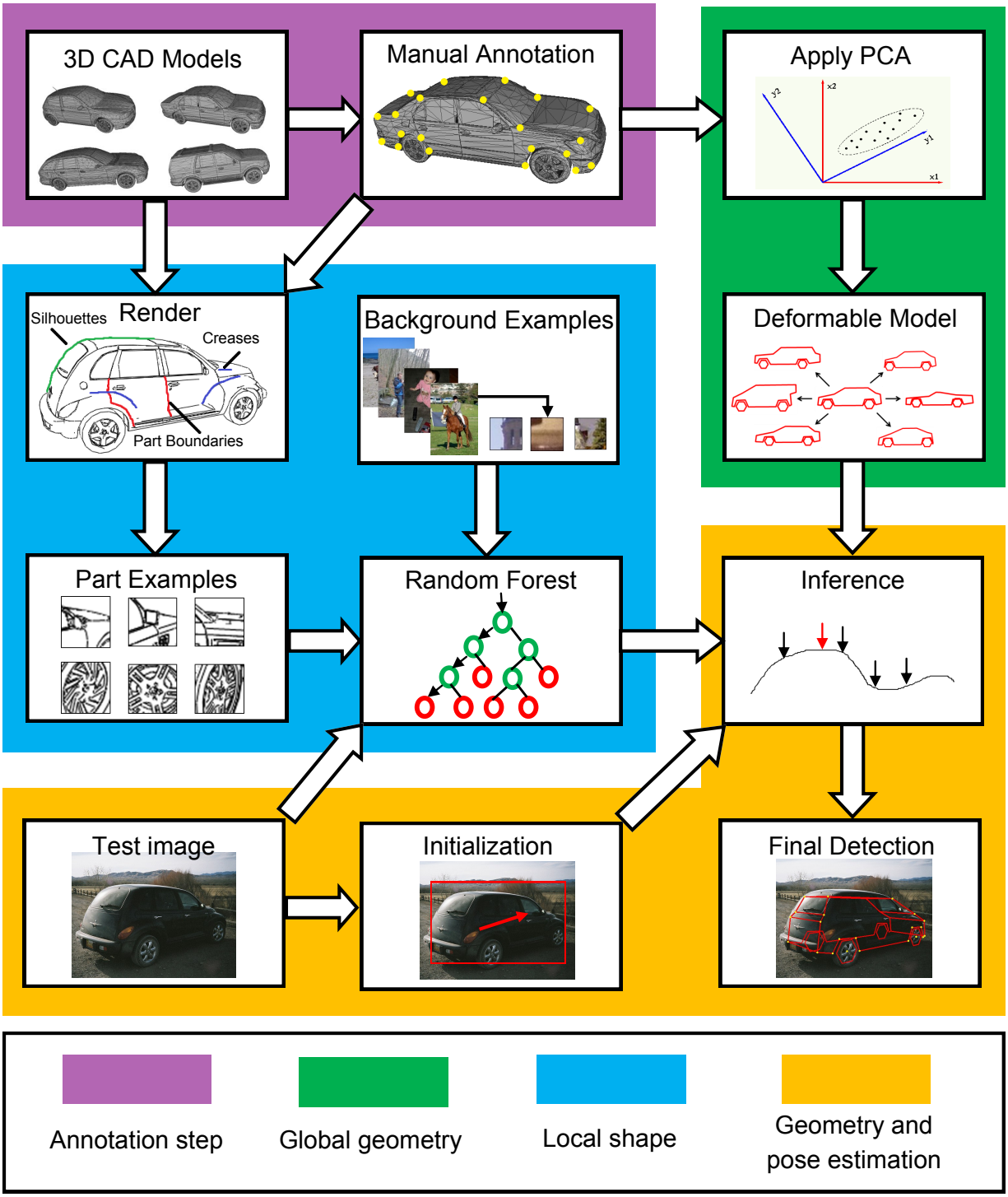


Figure 3.2: Full system diagram.

to the test image. We then explore the space of possible object geometries and poses by guided random sampling from the shape model, in order to identify the ones that best agree with the image evidence. The system is schematically depicted in Figure 3.2.

Contributions. The paper makes the following contributions. (i) we show that for certain object types classical 3D geometric object class representations better fulfill

the requirements of detailed visual modeling, and deliver object hypotheses with much more geometric detail than current detectors (see Figure 3.1). We believe this geometric richness is an important ingredient for scene-level geometric reasoning. (ii) we demonstrate that a 3D model enriched with local appearance descriptors can accurately predict 3D object pose and shape from single still images. In particular, our model improves over state-of-the-art results for pose estimation on a standard multi-view dataset. (iii) we show the benefit of detailed geometric category models for a geometric modeling task, namely ultra-wide baseline matching, where we successfully recover relative camera pose over viewpoint changes up to 180°, again improving over previous work. And (iv) we give experimental results on predicting more fine-grained object categories (different types of cars and bicycles) based solely on the inferred 3D geometry.

Parts of this work have appeared in a preliminary conference paper (Zia et al., 2011). The present paper introduces an appearance model based on random forests which is both more accurate and much more efficient, a modified objective function for model-to-image matching, and improved and extended experimental results, including the addition of the challenging bicycle class.

The remainder of this paper is structured as follows. Section 3.3 reviews related work. Section 3.4 introduces our 3D geometric object class model. Section 3.5 gives experimental results, and Section 3.6 concludes the paper with an outlook on future work.

3.3 Related work

Our work attempts to recover detailed geometric 3D object representations from single input images. As such, it is related to 3D geometric modeling from the earlier days of computer vision, more recent advances in scene understanding, and multi-view object class recognition, each of which we review in the following.

Early 3D modeling. Geometric modeling in 3D used to be an important component of visual object recognition, from the inception of computer vision until about the mid 1990ies. Many systems (Roberts, 1963; Brooks, 1981; Pentland, 1986) were proposed which built complex shapes from simpler primitives, such as polyhedra (Roberts, 1963), generalized cylinders (Brooks, 1981), and super-quadrics (Pentland, 1986). With these primitives, single objects as well as entire scenes were represented. Alternatively, salient local parts of the 3D shape, such as triplets of line segments, were matched to their image projections (Lowe, 1987). Hand-crafted, rigid 3D models were proposed to track vehicles in scenes with static background (Haag and Nagel, 1999; Koller et al., 1993), later extended to deformable models (Sullivan et al., 1995).
Unfortunately, while these models provided rich descriptions of objects and scenes, robustly matching them to cluttered real-world images proved to be exceedingly difficult at the time. Thus, later research abandoned them in favor of less expressive, but more robust 2D models. These include sparse sampling at locally confined regions of interest (Agarwal and Roth, 2002; Csurka et al., 2004; Leibe et al., 2006); modeling the spatial relationship between these regions at different levels of detail (Fergus et al., 2003; Felzenszwalb et al., 2010), or not considering such relations at all (Csurka et al.,

2004); and densely sampling (usually gradient-based) features from the object's extent in 2D (Dalal and Triggs, 2005).

Recent 3D modeling. With the advent of powerful computers and advances in machine learning, it has become feasible to revisit some of the classical ideas of 3D object modeling. In the context of indoor scene understanding, Wang et al. (2010) proposes a method to infer the 3D layout of the walls and segment out the clutter objects, and Hedau et al. (2010) shows that such 3D modeling not only provides a better interpretation of the scene, but also improves 2D object detection performance. Along the same lines, Hoiem et al. (2008) models interactions between objects, surface orientations, and 3D camera viewpoint for outdoor scene understanding, and demonstrates improved performance in object detection. Gupta et al. (2010) takes into account qualitative geometric and mechanical properties of objects and model their relationships, in order to generate qualitative 3D interpretations of outdoor scenes. Similarly, pedestrian and vehicle tracking from mobile platforms has been demonstrated to benefit from 3D reasoning (Ess et al., 2009; Wojek et al., 2010).

Inspired by this comeback of 3D scene understanding, our work aims to furnish the underlying representations with a lot more geometric detail (Zia et al., 2011). By combining a deformable 3D shape model with powerful local descriptors, we obtain more detailed and more expressive object class models, that directly lend themselves to detailed 3D reasoning about object and scene geometry. Recent works with similar ambitions as ours are Xiang and Savarese (2012) and Hejrati and Ramanan (2012). An object is represented in Xiang and Savarese (2012) as a collection of a few planar segments in 3D space called "aspect parts" (*e.g.* one planar "aspect" for the bicycle class, six for the car class). Like us they train on 3D CAD models, manually defining the aspect parts for different object categories. Geometric relations are represented in a similar way as in Savarese and Fei-Fei (2007), whereas pose is represented by a discrete set of viewpoints. In Hejrati and Ramanan (2012), a 2D part-based object model predicts the location of land marks, which is lifted to 3D in a second stage by fitting a coarse 3D model to these land marks with non-rigid SfM. In our work, we go even further in terms of 3D detail and predict in a continuous pose space. In another paper Zia et al. (2013), we apply our representation to explicitly model occlusions.

Multi-view recognition. A closely related problem to ours is multi-view recognition, which has received a lot of interest in recent years. The most frequently used approach for that task are banks of viewpoint-specific detectors (Schneiderman and Kanade, 2000; Ozuysal et al., 2009; Zhu et al., 2010; Stark et al., 2010; Villamizar et al., 2011;Payet and Todorovic, 2011; Pepik et al., 2012b). Other approaches, while still relying on several flat, viewpoint-specific representations, establish connections between viewpoints via homographies (Yan et al., 2007), probabilistic morphing of object parts (Su et al., 2009), discriminative mixtures of global templates (Gu and Ren, 2010), or by feature tracking with integrated single-view codebooks (Thomas et al., 2006). One step further towards true 3D recognition are models with rigid 3D configurations of local 2D features (Liebelt et al., 2008; Arie-Nachimson and Basri, 2009; Glasner et al., 2011).

Similar to the renewed trend of 3D modeling on the scene-level, attempts are recently

being made to explicitly represent 3D object class geometry alongside appearance. A coarse, volumetric blob model is learned from 3D CAD data in Liebelt and Schmid (2010), and combined with 2D appearance models, which have been learned from annotated real-world images. The implicit shape model (Leibe et al., 2006) is augmented in Sun et al. (2010) with the relative depth between codebook entries, obtained from a structured light system. Pepik et al. (2012b) extend the deformable part model (DPM) of Felzenszwalb et al. (2010) to include coarse viewpoint estimates in a structured prediction framework, and enforce part correspondences across viewpoints by 3D constraints.

While these approaches internally capture 3D object class geometry to some degree, they typically still provide 2D bounding boxes and coarse viewpoint labels as their output, and do not guarantee that the local parts are localized correctly. In contrast, our method generates complete hypotheses of 3D object geometry, including continuous viewpoint estimates with 5 degrees of freedom.

Efficient part detection. As the number of object classes and viewpoints increases, the computational cost for appearance-based detection grows significantly. Several attempts have been made to solve this problem by sharing information between object classes on different levels, *e.g.* Salakhutdinov et al. (2011). Random Forests (Breiman, 2001) provide a natural way to perform classification with multiple classes, and allow sharing at the level of weak learners inside the algorithm. They have successfully been used to train detectors for interest points (Lepetit and Fua, 2006; Leistner, 2010). In our experience, random forests also handle multi-modal distributions rather well. We use a single multi-class random forest classifier with one class per object part, combining examples from many different viewpoints in each class.

3.4 3D Geometric object class model

Decomposing object class representations into separate components for global layout and local appearance is a widely accepted paradigm in object class recognition (Fergus et al., 2003; Felzenszwalb et al., 2010). Its main advantages are the ability to account for variations in object shape better than rigid template models, and robustness to partial occlusion. The paradigm is often implemented by optimizing a smooth, continuous function of the global layout at recognition time, *e.g.* in the form of tree-structured (Felzenszwalb et al., 2010) or fully connected (Fergus et al., 2003) Gaussian densities over part positions. While these approaches have efficient implementations and have proven robust in terms of image matching, the resulting object hypotheses are hard to interpret and reason about in terms of geometry: deviations from geometrically plausible layouts are merely penalized, but not rendered impossible, and in fact individual parts are misplaced rather frequently.

Since we aim to not only detect the object, but also recover its geometry, we choose a different route and generate only geometrically valid hypotheses to start with. In a second step, we then verify that the generated hypotheses are supported by sufficient image evidence, a strategy sometimes termed *hypothesize-and-verify*, or *sample-based inference*.

We model an object class as a 3D wireframe representing global layout, with attached local appearance representations of object parts. Like several other recent works in multi-view recognition we leverage synthetic training data besides real-world images (Liebelt and Schmid, 2010; Stark et al., 2010; Zia et al., 2011; Pepik et al., 2012b), and learn both shape and appearance from a collection of 3D computer aided design (CAD) models, thereby ensuring consistency between global layout and local part models by design. At recognition time, we establish the connection between the 3D wireframe and the 2D image by means of a projective transformation, which is part of the object hypothesis. The transformation could potentially be shared among multiple objects in the same scene, however this is not further explored here.

3.4.1 Global geometry representation and learning

Our global geometry representation is given by a deformable 3D wireframe, which we learn from a collection of exemplars obtained from 3D CAD models. More formally, a wireframe exemplar is defined as an ordered collection of n vertices, residing in 3D space, chosen from the set of vertices that make up a 3D CAD model. In our current implementation the topology of the wireframe is pre-defined (manually defined for each object class, similar to Xiang and Savarese (2012)) and its vertices are chosen manually on the 3D CAD models. In the future, they could potentially be obtained using part-aware mesh segmentation techniques from the computer graphics literature (Shalom et al., 2008).

We follow the classical "active shape model" formulation of point-based shape analysis (Cootes et al., 1995), and perform PCA on the resulting (centered and rescaled) vectors of 3D coordinates. The final geometry representation is then based on the mean wireframe μ plus the m principal component directions $\mathbf{p}_j$ and corresponding standard deviations σ_j, where $1 \leq j \leq m$. Any 3D wireframe $\mathbf{X}$ can thus be represented, up to some residual ϵ, as a linear combination of r principal components with geometry parameters $\mathbf{s}$, where s_k is the weight of the k^{th} principal component:

$$\mathbf{X}(\mathbf{s}) = \mu + \sum_{k=1}^{r} s_k \sigma_k \mathbf{p}_k + \epsilon \tag{3.1}$$

Example 3D wireframe models for cars and bicycles are shown in Figure 3.3. Please note how principal directions represent the diversification of cars into sedan, SUV, sports car, and compact car, and of bicycles into mountain bike, racing bike, and children's bike. In our experiments, we show that we can in fact recover these fine-grained vehicle categories by fitting the model to single input images (Section 3.5.6).

3.4.2 Local shape representation

In order to match the 3D geometry representation to real-world images, we train a distinct part shape detector for each vertex in the wireframe, for a variety of different viewpoints. This is in contrast to early approaches relying on the matching of discrete image edges to model segments (Koller et al., 1993; Sullivan et al., 1995; Haag and Nagel, 1999), which

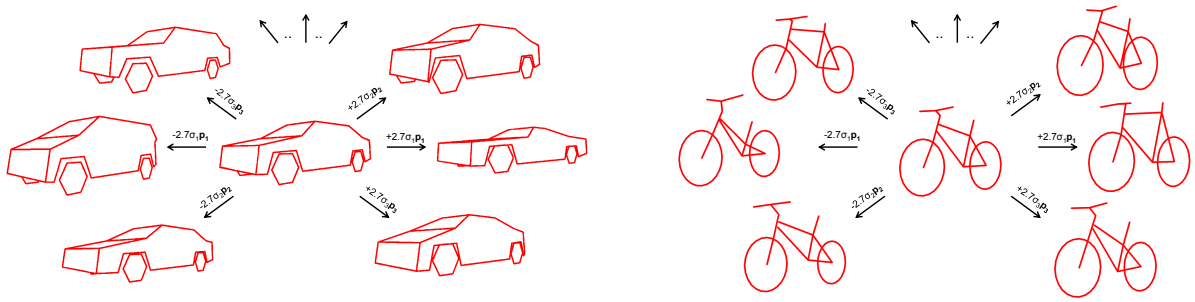


Figure 3.3: Coarse 3D wireframe representations of cars (left) and bicycles (right). Modes of variation along the first three principal component directions.

has proven to be of limited robustness in the face of real-world image noise and clutter.

Following Stark et al. (2010); Zia et al. (2011), we employ sliding-window detectors, searching over image locations and scales, using a dense variant of shape context (Andriluka et al., 2009) as features. For each wireframe vertex, a detector is trained from vertex-centered patches of non-photorealistic renderings of our 3D CAD models (Figure 3.4). Despite the apparent difference from real-world appearance, this particular combination of edge-based rendering and shape feature has shown to generalize well from rendered to real-world images (Stark et al., 2010; Pepik et al., 2012b). Rendering positive training examples further has the advantage of being able to generate massive amounts of artificial training data from arbitrary viewpoints. Following Stark et al. (2010); Zia et al. (2011); Pepik et al. (2012b), we render three different types of edges: crease edges, which are inherent properties of a 3D mesh, and thus invariant to the viewpoint, part boundaries, which mark the transition between semantically defined object parts and often coincide with creases, and silhouette edges, which describe the viewpoint-dependent visible outline. Negative training data is obtained by sampling random patches from a set of real-world background images set, as well as random patches from rendered images in the vicinity, but not on the parts of interest. The latter is important in order not to bias the part detectors to label all photorealistic patches as background, and also improves localization accuracy of the detectors.

3.4.3 Discriminative part detection

As local part detectors, we use discriminative classifiers trained for a discrete set of viewpoints, specified by azimuth and elevation angles. We explore two different variants, namely individual binary AdaBoost (Freund and Schapire, 1997) classifiers per part and viewpoint, and a monolithic multi-class random forest (Breiman, 2001) per object class. As we show in our experiments (Section 3.5.3), random forests prove favorable w.r.t. runtime while maintaining the same part localization performance, which is why all following results in Section 3.5 are based on random forests.

AdaBoost. In this variant, we train for each part and each viewpoint an individual binary AdaBoost classifier, which discriminates that particular part in that particular view from

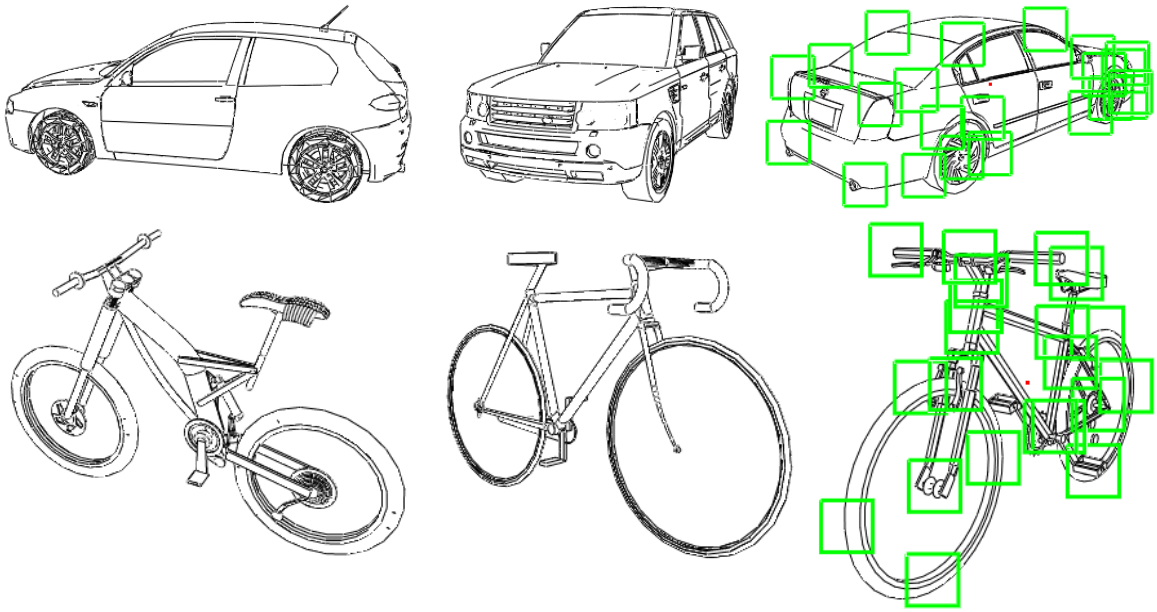
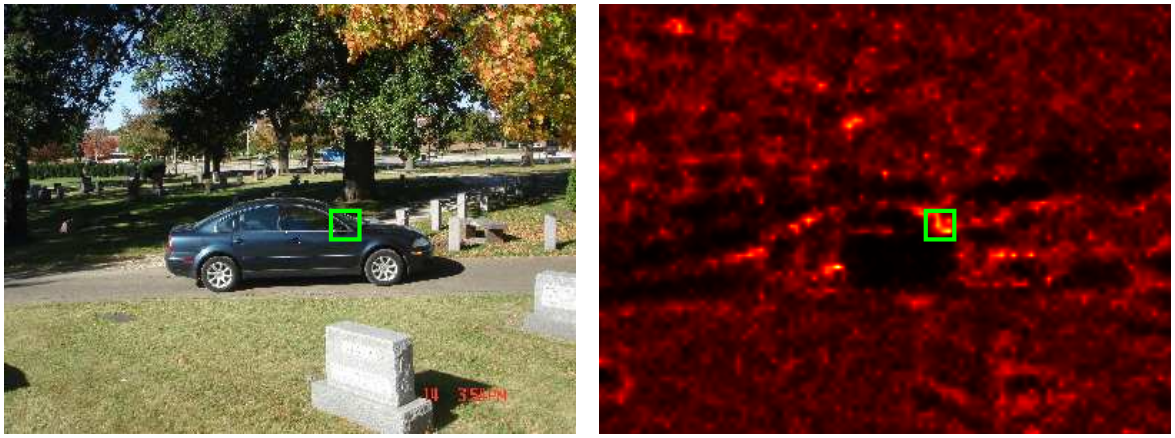


Figure 3.4: Non-photorealistic renderings for local part shape detector training, cars (top), bicycles (bottom). Green boxes denote positive training examples.



(a) (b)

Figure 3.5: Random forest detection map for one car part. (a) Test image and ground truth part, (b) detection map. Brighter shade corresponds to higher likelihood.

the background. Such a strategy has been employed successfully for people detection in Andriluka et al. (2009), and in our previous work (Zia et al., 2011).

Random forest. In an attempt to reduce the massive amount of detectors arising from the cross product of parts and viewpoints, we make two modifications to the above scheme. First, we replace the binary classifiers by a single multi-class classifier with one class per part (plus one for the background). We choose random forests (Breiman, 2001), since they have been shown to deliver excellent performance for multiclass problems with complex class-conditional distributions. Second, we leverage the ability of random forests

to model multi-modal distributions, and combine all training examples into a single class that depict the same part at any viewpoint. That is, we train a single viewpoint-invariant random forest, which distinguishes between parts, irrespective of the viewpoint.

In the individual nodes of the decision trees, we use *oblique* splits that decide based on random hyper-planes of a larger number of randomly chosen dimensions (Menze et al., 2011), as opposed to the more commonly used *axis-aligned* (or *orthogonal*) splits, where node decisions are based on a single feature dimension. Oblique splits increase the discriminative power in connection with high-dimensional features, such as our dense shape context features. Furthermore we use the ratio between the part-conditional distribution and the background as final part detection score, as in Fergus et al. (2003); Villamizar et al. (2011). Figure 3.5(b) gives a random forest detection map for the car part of Figure 3.5(a).
Our quantitative evaluation indicates that the detection maps from random forests, although more diffuse due to the marginalization over viewpoints, provide a better tradeoff between discrimination and recall when used in combination with the global geometry model (Section 3.5.3).

3.4.4 Viewpoint-invariant shape & pose estimation

During recognition, we seek to find an instance of our 3D geometric model that best explains the observed image evidence. This is formulated as an objective function defined over possible configurations of the model as well as its projection to the test image. It is worth noting that this entails a search over continuous 3D geometry and viewpoint parameters rather than switching or interpolating between flat viewpoint-dependent representations as in previous work (Thomas et al., 2006; Su et al., 2009; Stark et al., 2010; Pepik et al., 2012b).

More formally, we denote a recognition hypothesis as $\mathbf{h} = (\mathbf{s}, f, \boldsymbol{\theta}, \mathbf{q})$. It comprises object geometry parameters $\mathbf{s}$ (see Section 3.4.1), camera focal length f, spherical viewpoint parameters for azimuth and elevation $\boldsymbol{\theta} = (\theta_{az}, \theta_{el})$, and image space translation and scale parameters $\mathbf{q} = (q_x, q_y, q_s)$. For perspective projection we assume a simplified projection matrix P that depends only on f, $\boldsymbol{\theta}$, and $\mathbf{q}$. It is composed of a camera calibration matrix $\mathsf{K}(f)$ and a rotation matrix $\mathsf{R}(\boldsymbol{\theta})$, and projects wireframe vertices $\mathbf{X}_j(\mathbf{s})$ to image coordinates $\mathbf{x}_j$:

$$\begin{aligned} &\mathsf{P}(f, \boldsymbol{\theta}, \mathbf{q}) = \mathsf{K}(f) \begin{bmatrix} \mathsf{R}(\boldsymbol{\theta}) & -\mathsf{R}(\boldsymbol{\theta})\mathbf{q} \end{bmatrix} \\ &\mathbf{x}_j = \mathsf{P}\mathbf{X}_j(\mathbf{s}) \ . \end{aligned} \tag{3.2}$$

For recognition, we want to find the maximum a-posteriori estimate

$$\hat{\mathbf{h}} = \arg\max_h \left[\mathcal{L}(\mathbf{h}) + \lambda \mathcal{Q}(\mathbf{h}) \right] \ , \tag{3.3}$$

where $\mathcal{L}(\mathbf{h})$ is the data likelihood term and $\mathcal{Q}(\mathbf{h})$ is a shape prior (regularizer).

Data likelihood and shape prior. The inference in our framework (see below) is based on sampling part configurations from the explicit 3D model (3.1) and scoring them. In such a model-driven approach only globally plausible shapes are ever generated, which

allows for a relatively simple data likelihood (compared to approaches where the part locations can move independently in a data-driven manner (Stark et al., 2010)).

We define the (log-)likelihood of an object instance being present as a sum over the likelihoods of its constituent parts, assuming conditional independence between them. The likelihood $S_j(\boldsymbol{\varsigma}, \mathbf{x}_j)$ of part j being present at any given image location $\mathbf{x}_j$ and local scale $\boldsymbol{\varsigma}$ has already been estimated by the part detector (Section 3.4.3). Following Villamizar et al. (2011) we normalize the part likelihood by the background likelihood $S_b(\boldsymbol{\varsigma}, \mathbf{x}_j)$ at the same location. In order to account for object-level self-occlusion, only parts that are visible in the putative projection are considered, leading to binary indicator functions $o_j(\mathbf{s}, \boldsymbol{\theta})$ for the visibility. Finally the likelihood is re-normalized to the number of visible parts. The complete data term then reads

$$\mathcal{L}(\mathbf{h}) = \max_{\boldsymbol{\varsigma}} \left[\frac{1}{\sum_{j=1}^{m} o_j(\mathbf{s}, \boldsymbol{\theta})} \sum_{j=1}^{m} o_j(\mathbf{s}, \boldsymbol{\theta}) \; log \frac{S_j\big(\boldsymbol{\varsigma}, \mathrm{P}\mathbf{X}_j(\mathbf{s})\big)}{S_b\big(\boldsymbol{\varsigma}, \mathrm{P}\mathbf{X}_j(\mathbf{s})\big)} \right] \tag{3.4}$$

The PCA model (3.1) implies a zero-mean multivariate Gaussian distribution of the shape parameters around the mean shape. Consequently we introduce a shape prior which penalizes deviations from the mean 3D shape of the object class according to

$$\mathcal{Q}(\mathbf{h}) = \sum_{k=1}^{r} log \, \mathcal{N}(s_k; 0, 1). \tag{3.5}$$

To avoid overly unlikely shape hypotheses from the extreme tails of the Gaussian we limit the shape parameters to the range $|s_k| < 3$, such that they cover 99.7% of the shape variation observed in the training set.

Inference. The objective (3.3) cannot be easily maximized, since the data term is highly non-convex and—due to the binary $o_j(\mathbf{s}, \boldsymbol{\theta})$—also not smooth. We thus resort to a stochastic hill-climbing method. To account for the multi-modality of the posterior we generate multiple starting points ("particles") $\{\mathbf{h}_m^n\}$ with corresponding objective values $L(\mathbf{h}_m^n) + \lambda \mathcal{Q}(\mathbf{h}_m^n)$, and iteratively improve them through stochastic search. Each particle $\mathbf{h}_m^n$ corresponds to a distinct set of values in the space of object hypotheses $\{\mathbf{s}, \boldsymbol{\theta}, \mathbf{q}\}$, with m being the particle index and n the iteration.[1] The initial set of particles is drawn from a uniform distribution for the unknown shape parameters, whereas the parameters for location and pose are based on the initialization. In every iteration the particles are then updated to increase their objective value (3.3). Instead of computing gradients, semi-local update steps are determined by random sampling, which copes better with weak local minima and avoids problems due to visibility changes: for each particle a number of candidates $\{\tilde{\mathbf{h}}_m^{n+1}\}$ are generated by drawing new values for the individual parameters h_m from Gaussians centred at the current values,

$$\tilde{h}_m^{n+1} \sim p(\tilde{h}_m^{n+1} | h_m^n) = \mathcal{N}\big(h_m^n, \sigma_h^2(n)\big) \; . \tag{3.6}$$

Among the candidates the one with the highest likelihood replaces the original particle,

[1] f is held fixed in our experiments, assuming that the perspective effects are similar for all images.

thus yielding a new particle set $\{\mathbf{h}_m^{n+1}\}$. The variances $\sigma_h^2(n)$ of the proposal distributions are successively reduced according to an annealing schedule, for faster convergence. After the last iteration the particle with the highest weight is kept as MAP-solution $\hat{\mathbf{h}}$. Although the underlying posterior distribution may be very complicated, hill-climbing with simple Gaussian perturbations works well in practice. This procedure is similar to Leordeanu and Hebert (2008) (per particle), except that instead of computing the variances as a function of drawn samples, we choose them according to a pre-defined schedule. While this means that each of our particles might get stuck at local optima, keeping of multiple particles allows choosing the best one among them as well as keeping extra locally optimal hypotheses for a future scene-level reasoning stage.

Initialization. Rather than running inference blindly over entire test images, we start from promising image positions, scales, and viewpoints, which we obtain in the form of predicted object bounding boxes from a conventional 2D multi-view detector. In particular, we use the recently proposed multi-view extension of the deformable part model by Pepik et al. (2012b), which has been shown to yield excellent performance w.r.t. both 2D bounding box localization and coarse viewpoint classification. Specifically, we initialize q_x and q_y inside of a predicted object bounding box, and q_s according to the bounding box size. Similarly, we initialize the viewpoint parameters $\boldsymbol{\theta}$ according to the coarse viewpoint predicted by the detector. Due to the highly non-convex nature of the problem the overall system performance is strongly influenced by the initialization quality (Section 3.5.2).

3.5 Experimental evaluation

In the following, we carefully analyze the performance of our 3D object class model in a series of experiments, focusing on its ability to provide detailed 3D object geometry. To that end, we evaluate its performance in four different tasks, comparing to results reported by prior work where appropriate.

(i) first we evaluate the ability to accurately predict the locations of individual object parts in the 2D image plane (Section 3.5.3). In the context of 3D scene understanding, this ability is important in order to establish geometric relations between different scene entities, such as an object touching the ground plane at a specific location. (ii) we evaluate the ability to recover the full 3D pose of recognized objects (Section 3.5.4). In contrast to most prior work, we report results for both coarse *viewpoint classification* and continuous 3D *pose estimation* with 5 degrees of freedom (pictures are assumed to be upright, without in-plane rotation). In either case, we achieve results on par with or better than previous work. (iii) we evaluate our object class representation in the context of a 3D scene modeling task, namely to recover relative camera pose from wide-baseline pairs of images depicting the same object (Section 3.5.5). Here, the model is challenged to recover consistent 3D object geometries across different viewpoints, and improves over previously reported results for all baselines, up to 180°. (iv) we leverage the detailed 3D shape hypotheses provided by our approach for fine-grained object categorization based on geometric shape (Section 3.5.6).

3.5.1 Setup

We commence by describing the experimental setup w.r.t. test and training data, random forest training, inference, and initialization.

Test datasets. The evaluation is based on the *3D Object Classes* (Savarese and Fei-Fei, 2007) and *EPFL Multi-view cars* (Ozuysal et al., 2009) datasets, which both have been designed specifically for multi-view recognition. These datasets constitute a suitable trade-off between controlled conditions for experimentation and challenging real-world imagery. Our focus is on the object classes *car* and *bicycle*. The *3D Object Classes* test set depicts 5 object instances from 8 different azimuth angles, 3 distances, and 2 (cars) or 3 (bicycles) elevation angles, against varying backgrounds, amounting to a total of 240 cars and 360 bicycle test images. The *EPFL Multi-view cars* test set comprises 10 different car models with largely varying shape, rotating on a platform, with a sample every 3 to 4 degrees, totaling to about 1000 images. Figure 3.10 and 3.11 show qualitative results obtained by our method on images of these data sets.

Synthetic training data. In all experiments, we use 38 commercially available 3D CAD models of cars[2] and 32 freely available CAD models of bicycles[3] for training. We annotate 36 model points for cars and 21 for bicycles (Figure 3.4) in order to train both global geometry (Section 3.4.1) and local part shape (Section 3.4.2). Each part is rendered from 72 different azimuth ($5°$ steps) and 2 elevation angles ($7.5°$ and $15°$ above the ground) for cars, respectively 3 elevation angles ($7.5°$, $15°$, and $30°$) for bicycles, densely covering the relevant part of the viewing sphere (the bicycle test set covers a larger range of viewpoints). CAD models are rendered using the non-photorealistic style of (Stark et al., 2010; Pepik et al., 2012b). Rendered part patches serve as positive examples, randomly sampled image patches as well as non-part samples from the renderings serve as negative examples. The total number of training patches is 140,000 per class, evenly split into positive and negative ones.

Random forest training. As part detectors, we train a single random forest classifier (Breiman, 2001) for each object class (one for bicycles and one for cars), distinguishing between the parts of interest (36 for cars, 21 for bicycles) and background. In both cases the random forests have 30 trees with a maximum depth of 13. Node tests are given by random hyperplanes of dimensionality 59 (chosen from a total of $3,500$ dimensions of the shape context descriptor), which for our high-dimensional input we found empirically to deliver much higher performance than the more commonly used single dimension node tests.

Inference. We sample θ_{az} over a continuous range of $20°$ centered around the initialization and θ_{el} from ground level to 20° for cars and $30°$ for bicycles. For part detections, we consider the maximum score in a scale range of $\pm$30% around the bounding box scale.

Initialization. We report results for two different, informed initializations of our model, as well as results obtained by running our model from random starting points, not using any

[2] www.doschdesign.com

[3] www.sketchup.google.com/3dwarehouse/

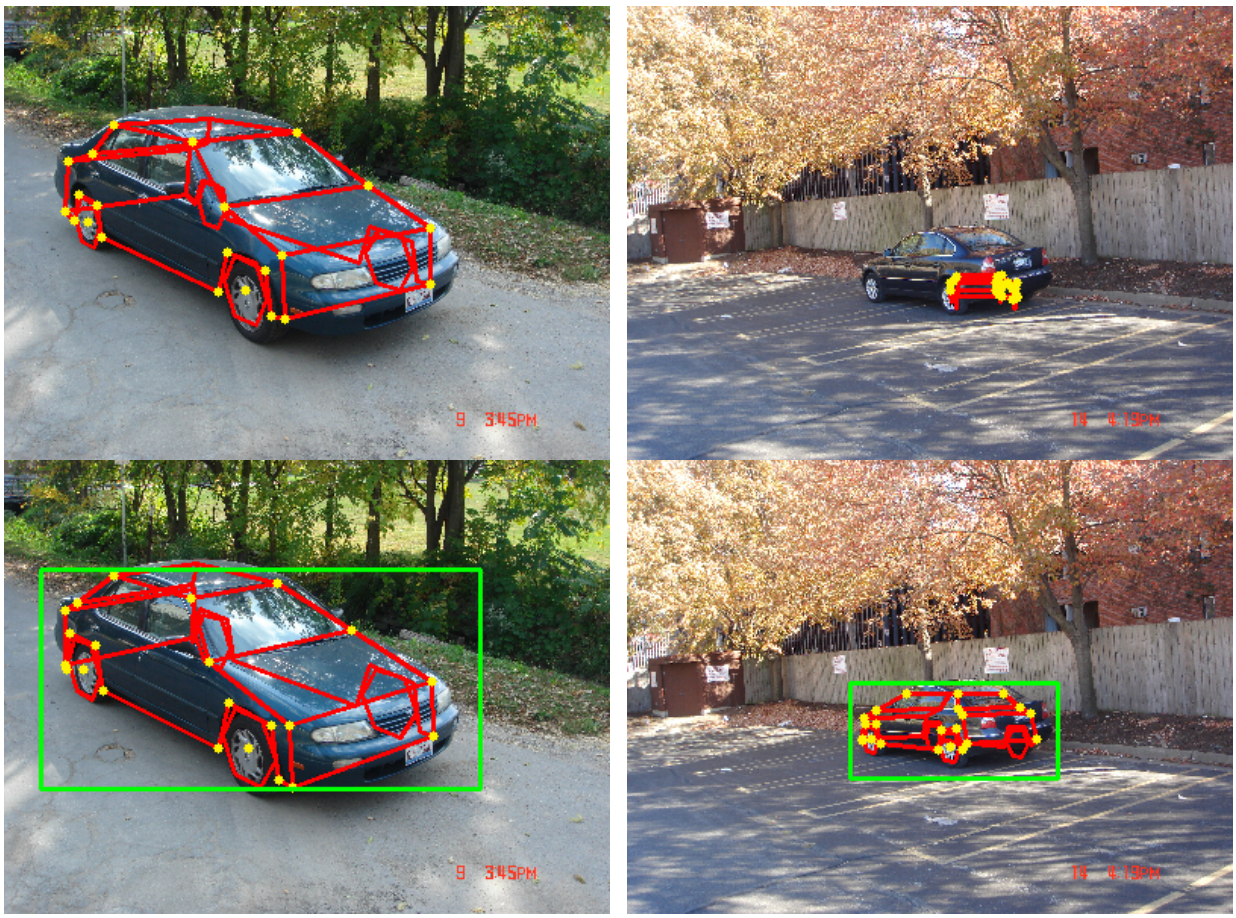


Figure 3.6: Example detections without (top row) and with informed initialization Pepik et al. (2012b) (bottom row).

prior information about object location and pose (Section 3.5.2).

The first initialization is provided by the state-of-the-art multi-view detector (Pepik et al., 2012b), providing almost perfect 2D bounding box localization on the *3D Object Classes* dataset for cars and bicycles (97.5% average precision each). Specifically, we use the multi-view DPM referred to as DPM-VOC-VP in Pepik et al. (2012b), trained from the respective car and bicycle training sets provided by the *3D Object Classes* and *EPFL Multi-view cars* data sets (Ozuysal et al., 2009). In the following, we refer to the combination of this initialization and our model as the *full system*, since it constitutes a fully automatic procedure that infers detailed 3D geometric hypotheses from input images, as it would be used in a real-world application.

The second initialization (termed *GT*) aims at providing a best case evaluation of our model isolated from the effects of the multi-view DPM, starting from annotated ground truth bounding boxes and coarse viewpoint estimates.

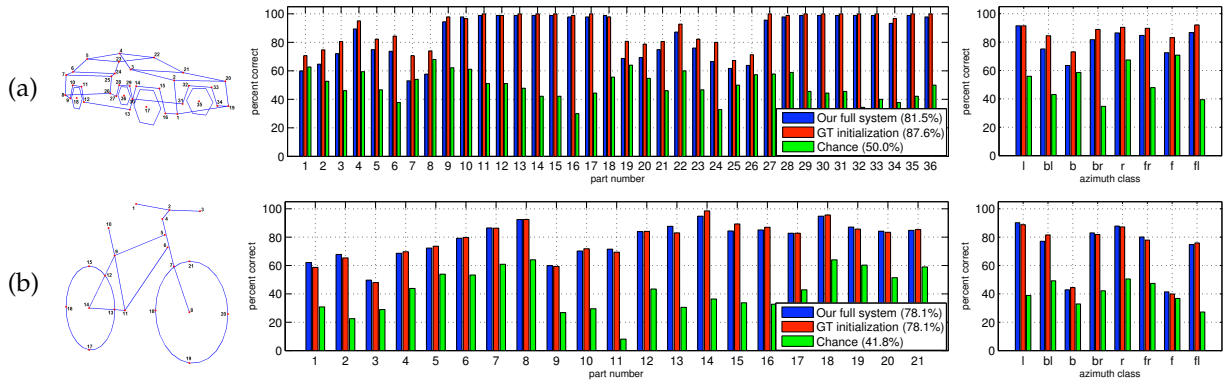


Figure 3.7: Part localization results on *3D Object Classes*. Part numbering schemes (left), localization performance for individual parts (center) and viewpoints (right), for (a) bicycles, and (b) cars.

3.5.2 Recognition without initialization

We commence by exploring the performance of our approach in isolation, independent from any informed initialization, by running it from a number of randomly selected starting points (250 particles drawn uniformly at random from the location, pose, and shape parameter space). We evaluate over the car class in the *3D Object Classes* dataset. Considering the highest scoring hypothesis in each of the 240 test images, we are able to localize the correct 2D bounding box in 51.7% of the cases (according to Pascal criterion (Everingham et al., 2010)). Further, following the experimental protocol of Su et al. (2009), the viewpoint of these true positive detections is correctly classified into 8 different azimuth angle classes (*left, front-left, front, front-right, right, back-right, back, back-left*) in 66.9% of the cases.

Although these numbers are encouraging, running our detailed 3D geometric model blindly over entire test images is obviously inferior to current state-of-the-art object class detectors, both w.r.t. 2D localization performance and computational complexity. In the following, we thus provide our model with *informed initializations* in the form of rough 2D object locations and poses (*full system*), obtained by a 2D multi-view detector (Pepik et al., 2012b). This cascaded approach drastically reduces computation, and results in state-of-the-art performance in pose estimation (Section 3.5.4). Figure 3.6 compares example detections obtained with and without informed initialization.

Please note that other recent work (Li et al., 2011) on deformable object models also relies on initializing models within a small operating window centered around the object (much like our *GT* initialization), and even assumes fixed object scale.

3.5.3 Part localization

One way of performing accurate geometric reasoning on the scene-level is to have object class models that provide well-defined anchor points, so as to geometrically relate them to other scene entities. Consider for example the wheels of a vehicle, which can be assumed to rest on a supporting surface, and can hence provide hints on the likely

position and orientation of a ground plane. Likewise, localizing extremal points on the vehicle body (such as bumper corners) can help to assess the area of covered ground and hence its 3D extent in the scene.

Since the parts in our model are chosen to correspond to well-defined regions of an object's anatomy (Section 3.4.1), we can evaluate the ability of our model to localize these parts individually. To that end, we annotate the 2D locations of all visible parts in our test images. We have made all annotations publicly available[4].

Protocol. We measure part localization accuracy as the fraction of correctly localized parts of a specific type across test images, restricted to those test images where the method under consideration delivers a true positive detection in terms of the Pascal criterion (Everingham et al., 2010) on the 2D bounding box. A part is considered correctly localized if its estimated 2D position deviates less than a fixed number of pixels from annotated ground truth, relative to its estimated scale. For instance, for a car side view at scale 1.0, covering 460×155 pixels, that number is 20, which amounts to localizing a part to within $\approx 4\%$ of the car length. The same criteria is used for bicycles. Note that this strict criterion is applied in all cases, even for hypotheses with grossly wrong viewpoint estimates.

Results. Figure 3.7 gives the results for part localization for cars (a) and bicycles (b), averaged over all test images, grouped by individual parts (center bar plots) and viewpoints (right bar plots). We distinguish among the performance of the *full system* (blue bars), our system initialized from ground truth (*GT*,red bars), and a baseline also initialized from *GT*, but using uniform part score maps (*chance*, green bars).

Per-part evaluation. In Figure 3.7 (a) and (b) (center), we observe that there are in fact differences in the localization accuracy of different parts. Notably for cars (Figure 3.7(a)(center)), parts located in the wheel regions (9-18, 27-36) are localized almost perfectly by both the *full system* and when starting from *GT*. This is not surprising, since wheels provide plenty of local structure that can robustly matched by local part detectors, providing strong guidance for the geometric model. Parts on the front roof (4, 22) can also be localized with great accuracy (89.4% and 87.2% by the *full system*), followed by back roof parts (5 with 74.9% and 23 with 76.0%) and hood parts (3 with 72.1% and 21 with 74.9%). Parts in the trunk region tend to perform worse (7 with 53.0% and 25 with 61.7%). We attribute this difference to the greater flexibility that our learned global geometry model allows in the back: the collection of training CAD models comprises limousines and sports cars as well as SUVs and station wagons.

Bicycles (Figure 3.7(b)(center)) appear to be more challenging than cars in general (*GT* performance drops by 9.5% from 87.6% to 78.1%), possibly due to their wire-like nature, which amplifies the influence of background clutter. Concerning the ranking of the parts, we observe a similar trend as for cars: parts located on the wheels (7-8,12-21) have localization accuracy of at least 82.8% for the *full system*, whereas the wheel centers (8,14) even reach 92.4% and 94.8%, respectively. Again, parts that exhibit more

[4]http://www.igp.ethz.ch/photogrammetry/downloads

Classifier type	classifiers trained	class. tested per detection	1 mode	3 modes	post inference
AdaBoost (Zia et al., 2011)	5,184	36	56.3%	75.5%	79.9%
AdaBoost (Zia et al., 2011)	5,184	432	**61.4%**	**79.0%**	81.1%
Random forest	36	36	35.0%	57.8%	**81.5%**

Table 3.1: Comparison of part detector performance using random forests and AdaBoost (Zia et al., 2011) (on cars).

variability in the training CAD models perform worse, such as the handle region (1-3, between 49.7% and 67.8%) and the joint below the seat (9 with 59.5%).

On average, we achieve correct part localization in an encouraging 81.5% of all cases for cars using the *full system* (87.6% using *GT*), and in 78.1% for bicycles (for both *full system* and *GT*).

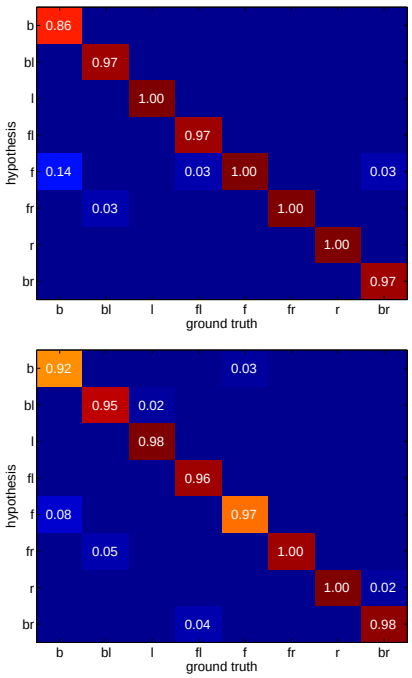
Per-viewpoint evaluation. Figure 3.7 (a) and (b) (right) groups the part localization results according to the different azimuth angles of test images, averaged over all parts. For cars (Figure 3.7(a)(right)), we observe that part localization performs best for plain side views (left 91.5%, right 86.5%, *full system*), followed by diagonal front (front-left 86.7%, front-right 84.7%) and back views (back-right 81.7%, back-left 75.2%). Plain front (72.5%) and back (63.5%) views perform moderately, apparently due to the absence of the strong evidence provided by the wheels in the other views.

The same tendency can be observed for bicycles (Figure 3.7(b)(right)). Plain side views perform best (left 90.2%, right 87.8%, *full system*), followed by the diagonal views (back-right 83.0%, front-right 80.1%, back-left 77.1%, front-left 74.8%) and the plain back and front views (42.9% and 41.4%).

Comparison to AdaBoost (Zia et al., 2011). Table 3.1 compares the part localization performance of the *full system* using *random forest* classifiers as part detectors with two variations of AdaBoost, as we previously proposed in Zia et al. (2011). The first variant trains a single binary AdaBoost classifier for each part (36 for cars), azimuth (72), and elevation angle (2), resulting in $5,184$ trained classifiers. At test time, only those classifiers belonging to the coarse viewpoint predicted by the initialization are considered (36 in total). The second variant uses the same set of trained classifiers, but considers neighboring viewpoints at test time as well (432 in total) to account for viewpoint uncertainty.
Table 3.1 gives results for post-inference part localization (*i.e.* , applying the *full system* end to end) as well as pre-inference localization, considering 1 and 3 highest modes in the part detection maps as hypotheses, respectively. When using 3 highest modes we consider a part detection as correct if any one of the modes falls on the ground truth part location. Not surprisingly, we observe that both AdaBoost versions perform much better in pre-inference localization than *random forests* (up to 26.4% for 1 and 21.2% for 3 modes), since the restriction to a narrow range of viewpoints increases the discriminative power of the resulting classifiers. While the inclusion of neighboring viewpoints aids

3D Object Classes	cars	bicycles
Liebelt et al. [34]	70.0%	75.5%
Stark et al. [51]	81.0%	-
Zia et al. [66]	84.0%	-
Glasner et al. [19]	85.3%	-
Payet et al. [43]	86.1%	80.8%
Initialization [45]	**97.5%**	**97.5%**
Full system	97.1%	97.1%
GT	98.7%	99.4%

(a)



(b)

Figure 3.8: Coarse viewpoint classification on *3D Object Classes*. (a) Accuracies, (b) confusion matrices for cars (top), bikes (bottom), using our *full system*.

robustness, including all viewpoints (as we do for *random forests*) degrades performance. Post-inference, however, *random forests* have a slight edge (81.5% vs. 79.9% and 81.1%), achieved with two orders of magnitude fewer classifiers (36 vs. $5,184$). This seemingly counter-intuitive behavior stems from the fact that in difficult cases the binary AdaBoost classifiers are sometimes "too convinced" that a part is *not* present, and these false negatives (low part likelihoods at the correct position) drive the inference away from the correct shape.

Summary. We conclude that our model yields accurate estimates of the 2D locations of individual parts in the majority of cases, providing a solid basis for 3D geometric reasoning. Since we also observe a non-negligible difference between the results obtained by different initializations (*full system* vs. *GT*), we expect further improvements in response to improved initial detections to start from.

3.5.4 Pose estimation

In this section, we evaluate the ability of our model to accurately estimate the 3D pose of recognized objects. Even without considering individual parts (as in Section 3.5.3), pose estimation facilitates monocular 3D perception and can provide valuable geometric information for scene-level reasoning. As an example, consider the effect of observing an object, say, a bicycle from different azimuth angles: knowledge about its 3D shape enables the viewer to estimate the perspective distortion not only of the object itself, but

EPFL Multi-view	cars
Ozuysal et al. (2009)	41.6%
Xiang and Savarese (2012)	64.8%
Lopez-Sastre et al. (2011)	66.0%
Initialization (Pepik et al., 2012b)	**76.5%**
Full system	**76.5%**

Table 3.2: Coarse viewpoint classification on *EPFL Multi-view cars*, using our *full system*.

of the entire scene, and thus reason about distances and relations in 3D Euclidean space. While the focus of our approach lies on providing detailed, continuous 3D pose estimates with 5 degrees of freedom (or even 6, if initialized with an object detector that is invariant to in-plane rotation), we start by reporting results for the popular task of viewpoint classification with 8 and 16 equally spaced viewpoint bins on *3D Object Classes* and *EPFL Multi-view cars*, respectively. In that setting, pose estimation is discretized into a multi-class labeling problem. Since our method relies on coarse viewpoint estimates provided by Pepik et al. (2012b) as an initialization, this evaluation also serves as a sanity check, to ensure that the added expressiveness of our model does not significantly degrade viewpoint classification performance.

Coarse viewpoint classification. Following the experimental protocol of Su et al. (2009), we report results on *3D Object Classes* dataset for the classification of true positive object detections according to 8 different azimuth angle classes (*left, front-left, front, front-right, right, back-right, back, back-left*). Figure 3.8(a) gives the corresponding results for cars and bicycles, comparing our *full system* to our system initialized from *GT* bounding boxes, the estimate provided by the *initialization* (Pepik et al., 2012b), and results reported in prior work.

In Figure 3.8(a), we observe that, for both cars and bicycles, the *initialization* (Pepik et al., 2012b) alone already provides almost perfect viewpoint classification (97.5% and 97.5%, respectively), outperforming the next best prior results (Payet and Todorovic, 2011) by margins of 11% and 17%, respectively. While our *full system* maintains that high level of performance for both classes (97.1% and 97.1%; compared to Pepik et al. (2012b) we mis-classify the viewpoint of a single car/bicycle), our model initialized from *GT* can further improve to 98.7% for cars and 99.4% for bicycles.

For the *EPFL Multi-view cars* dataset, we perform viewpoint classification into 16 azimuth angle classes as in Ozuysal et al. (2009). The test set contains 10 different car models imaged under fairly poor lighting conditions, thus the performance of most state-of-the-art methods is worse than the results over *3D Object Classes*, as indicated in Table 3.2. Again, the *initialization* (Pepik et al., 2012b) already obtains the best viewpoint classification accuracy reported to date. The *full system* again maintains the high level of classification accuracy (76.5 % for both *initialization* and *full system*), though it loses the detections on 9 test images out of 994.

3D Object Classes cars	Total Images	True Positives	% Correct Azimuth	Avg. Error Azimuth	Avg. Error Elevation
Stark et al. (2010)	48	46	67.4%	4.2°	4.0°
Zia et al. (2011)	48	45	73.3%	3.8°	**3.6°**
Initialization (Pepik et al., 2012b)	48	**48**	70.8%	**3.4°**	-
Full system	48	47	**95.7%**	3.8°	3.7°
Without init.	48	21	**61.9%**	3.9°	4.8°
GT	48	47	93.6%	3.6°	3.2°

(a)

3D Object Classes bicycles	Total Images	True Positives	% Correct Azimuth	Avg. Error Azimuth	Avg. Error Elevation
Initialization (Pepik et al., 2012b)	72	**69**	76.8%	**2.3°**	-
Full system	72	67	**89.6%**	3.4°	**10.4°**
GT	72	69	98.6%	3.2°	8.7°

(b)

Table 3.3: Continuous viewpoint estimation: (a) cars, (b) bicycles.

EPFL Multi-view cars	Total Images	True Positives	% Correct Azimuth	Avg. Error Azimuth
Initialization (Pepik et al., 2012b)	994	**981**	73.3%	3.4°
Full system	994	972	**80.3%**	**3.3°**

Table 3.4: Continuous viewpoint estimation (*EPFL cars*).

Continuous viewpoint estimation. Since the ground truth of the *3D Object Classes* dataset does not provide accurate viewpoints beyond the eight rough directions, we annotate all images depicting one particular car (48 images) and one particular bicycle (72 images) with continuous azimuth and elevation angles, by manually fitting 3D CAD models to the images. In particular, we start from a CAD model of maximally similar shape, placed on a ground plane, and iteratively adjust the 3D position of the car, the position and orientation of the camera, and its focal length. This procedure is quite time-consuming, but results in precise geometric fits for all images.

Table 3.3(a) and (b) give the results for continuous viewpoint estimation, comparing the *full system*, *GT*, and the *initialization* (Pepik et al., 2012b), again considering only final true positive detections. For cars (Table 3.3(a)), we also include previously reported results of Stark et al. (2010); Zia et al. (2011). A viewpoint estimate is considered correct if it lies within 10° of the annotated ground truth azimuth angle (in contrast to the 45° bins of coarse viewpoint classification). Among those correct estimates, we further measure and report the average angular error in both azimuth and elevation.

In Table 3.3(a), we observe that our *full system* improves by a remarkable 22.4% over our previous result of 73.3% (Zia et al., 2011) for cars, amounting to 95.7% viewpoint estimates that are within 10° of the ground truth. At the same time, we improve 24.9% over the *initialization* (Pepik et al., 2012b), confirming the ability of our method to provide

Azimuth Diff.	Image Pairs	SIFT Lowe (2004)	Parts only	Zia Zia et al. (2011)	DPM-3D-Const. Pepik et al. (2012b)	*Full system*	*GT*
45°	53	2.0%	30.2%	54.7%	54.7%	**86.8%**	86.8%
90°	35	0.0%	22.8%	60.0%	51.4%	**88.6%**	94.3%
135°	29	0.0%	20.7%	51.7%	51.7%	**65.5%**	89.7%
180°	17	0.0%	0.0%	41.2%	70.6%	**76.5%**	76.5%
Avg.	134	0.5%	18.4%	51.9%	57.1%	**79.4%**	86.8%

Table 3.5: Ultra-wide baseline matching results (cars).

viewpoint estimates of much finer detail than captured by coarse viewpoint classification. For bicycles (Table 3.3(b)), the improvement over the *initialization* (Pepik et al., 2012b) is less pronounced, but still significant (by 12.8% from 76.8% to 89.6%).

Among the correct viewpoint estimates, the actual viewpoint errors for cars are all in the same range. Our *full system* achieves angular errors of 3.8° in azimuth and 3.7° in elevation, which is practically the same as our model starting from *GT* (3.6° and 3.2°). Similar or slightly larger errors are also obtained with competing methods, which however have significantly lower recall, meaning that the "more difficult" cases solved only by our model are nevertheless accurately estimated. Similarly, we achieve 3.4° in azimuth and 10.4° in elevation for bicycles. We attribute the significantly larger elevation errors to the fact that bicycles are largely planar, and thus their elevation angle is rather correlated with the shape (in particular the height-to-length ratio).

Table 3.4 gives the corresponding results for *EPFL Multi-view cars*. Here, cars are depicted from a wide variety of viewpoints sampled densely from the entire 360° viewing circle. In unison with the results on *3D Object Classes*, we improve over the *initialization* (Pepik et al., 2012b) by 7%, obtaining precise azimuth angle estimation in 80.3 % of the cases, whereas the average error in azimuth estimation decreases to 3.3°.

3.5.5 Ultra-wide baseline matching

While the experiments of Section 3.5.3 (part localization) and 3.5.4 (pose estimation) evaluate our approach from an object-centric perspective, the following experiment quantifies its ability to recover 3D camera and scene geometry. In particular, we consider the task of estimating relative camera pose from a pair of images depicting the same scene, *i.e.* epipolar geometry fitting. This task quickly gets very challenging as the baseline increases; the best invariant interest point descriptors like SIFT (Lowe, 2004) allow matching up to baselines of ≈ 30 degrees in orientation and a factor of ≈ 2 in scale. Only recently, Bao and Savarese (2011) have noted that semantic knowledge ("the scene contains a car somewhere") can provide additional constraints for solving the matching problem, increasing the range of feasible baselines. Their approach enforces consistency between 2D object detection bounding boxes and coarse pose estimates across views in a structure-from-motion framework.

In contrast, we leverage the ability of our approach to predict accurate object *part positions*, and use those directly as putative matches. The 3D model is fitted independently

to two input images, and the model vertices form the set of correspondences. Matching is thus no longer based on the local appearance around an isolated point, but on the overall fit of the object model. Note, this makes it possible to match even points which are fully occluded. In principle, relative camera pose could be obtained directly from the two object pose estimates. In practice this is not robust, since independent fitting will usually not find the exact same shape, and even in a generally correct fit some parts may be poorly localized, especially if the guessed focal length is inaccurate. Hence, we use corresponding model vertices as putative matches, and robustly fit fundamental matrices with standard RANSAC.

Protocol. As test data we have extracted 134 pairs of images from the car data set, for which the car was not moved w.r.t. the background. The restriction to stable background ensures the comparison is not unfairly biased against SIFT: straight-forward descriptor matching does not need model knowledge and can therefore also use matches on the background, whereas interest points on the cars themselves are rather hard to match because of specularities.

To assess the correctness of the fundamental matrices thus obtained, we manually label ground truth correspondences in all 134 images pairs, on the car as well as the background. A fit is deemed correct if the Sampson error (Hartley and Zisserman, 2004) for these points is < 20 pixels.

Results. In Table 3.5, we compare, for varying angular baselines (45°, 90°, 135°, 180°), the results obtained by our *full system* and *GT* to previously reported results (our previous method Zia et al. (2011) and the multi-view deformable part model with 3D constraints, DPM-3D-Const. (Pepik et al., 2012b)), and two baseline methods: (i) we find putative matches with SIFT (using the default options in Vedaldi and Fulkerson (2008)); and (ii) in order to assess whether the geometric model brings any benefit over the raw part detections it is based on, we perform non-maximum suppression on the scoremaps and obtain three modes per part in each of the two images. The permutations of these locations form the set of putative correspondences.

As expected, SIFT catastrophically fails (0.5% correctly estimated relative poses on average). Matching raw part detections works slightly better (18.4%), since the dedicated detectors search for a pre-trained part irrespective of the viewpoint, rather than comparing low-level appearance patterns. The DPM-3D-Const. (Pepik et al., 2012b) already outperforms our previous result of 51.9% (Zia et al., 2011), but is in turn superseded by a significant margin of 22.3% by our *full system* (79.4%). Note that even for 180° viewpoint spacing, 76.5% of the estimated epipolar geometries are correct, see examples in Figure 3.11(g).

3.5.6 Fine-grained categorization by 3D geometry

In addition to the popular task of distinguishing between basic-level categories (such as cat and dog), fine-grained categorization into sub-ordinate categories (such as sheep dog and Labrador) has received increasing attention in the vision literature lately (Nilsback and Zisserman, 2008; Yao et al., 2011; Farrell et al., 2011). It is deemed challenging due

(a)

Car cat.	1	2	3	4	5	Total
Full System	**65.9%**	**81.3%**	60.4%	70.8%	**60.4%**	**67.8%**
GT	55.3%	70.8%	**64.6%**	**75.0%**	56.2%	64.4%
Chance	38.9%	30.5%	30.5%	38.9%	38.9%	35.5%

(b)

Bicycle cat.	1	2	3	4	5	Total
Full System	**57.3%**	**62.5%**	**71.2%**	**75.0%**	65.1%	**66.1%**
GT	55.1%	60.9%	68.6%	71.0%	**68.6%**	64.8%
Chance	25.0%	28.1%	40.6%	25.0%	25.0%	28.7%

Table 3.6: Fine-grained categorization of (a) cars , (b) bicycles.

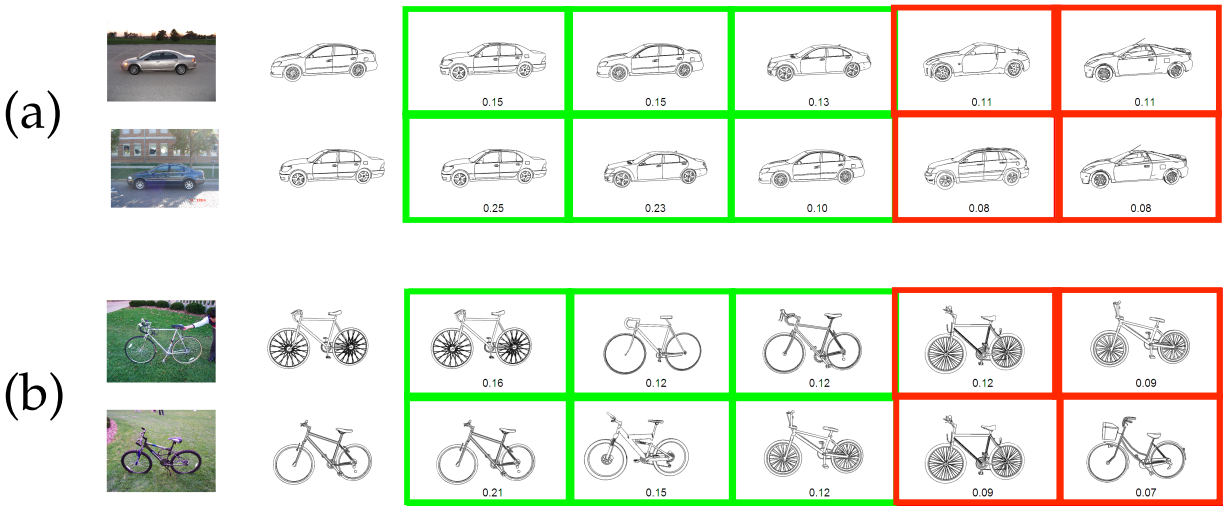


Figure 3.9: Fine-grained categorization examples for (a) cars, (b) bicycles. Example input image of true class with corresponding CAD model prototype (left), five most frequently matched CAD model hypotheses (right; green denotes correct, red incorrect matches).

to the need to capture subtle appearance differences between classes (*e.g.* , fur texture) while at the same time maintaining robustness to intra-class variations induced by viewpoint changes and lighting conditions. As a consequence, the focus has mostly been on classes and categorization methods that favor discrimination by strong local cues (such as random image patches (Yao et al., 2011; Farrell et al., 2011)) or global image statistics (such as color and gradient histograms for flowers (Nilsback and Zisserman, 2008)).

In the following experiment, we choose a different route, and base the fine-grained categorization entirely on 3D geometry. In particular, we consider the natural distinction between fine-grained sub-ordinate categories of cars and bicycles, such as sedans, sports cars, SUVs, etc. as well as mountain bikes, street bikes, etc.

We perform fine-grained categorization following a nearest neighbor scheme. Starting from a 3D wireframe estimate obtained by our model for a test image, we retrieve the closest wireframe exemplar from the database of CAD models of the basic-level object

class of interest (car or bicycle), using Euclidean distance between translation- and scale-invariant wireframe representations. Examples of nearest neighbor matches are visualized in Figure 3.11(a) - (f), which show edge renderings of retrieved CAD models, projected into the respective test image at the estimated location, scale, and viewpoint. Please note the remarkable accuracy of the fully automatic 3D geometry estimates.

Protocol. We suggest the following procedure to quantify the performance of fine-grained categorization based on the *3D Object Classes* data set: For each of the 5 car and 5 bicycle instances in the test set, we manually determine the single best matching CAD model in terms of 3D geometry, using the methodology described in Section 3.5.4. We then consider each of these CAD models a prototype of a fine-grained category, and measure how often the retrieved CAD models are sufficiently similar to these prototypes, by thresholding the mean Euclidean distance between corresponding vertices of the 3D fit and the annotated CAD model.

Results. Table 3.6 gives fine-grained categorization results for cars (a) and bicycles (b), comparing our *full system*, *GT*, and a *chance* baseline returning random CAD models from the database. The first five columns give the fraction of retrieved CAD models deemed sufficiently similar to the respective fine-grained category prototype. The last columns give the corresponding total fractions: for both cars (Table 3.6(a)) and bicycles (Table 3.6(b)), our *full system* successfully recovers the fine-grained category in two thirds of the cases (67.8% for cars, 66.1% for bicycles). Figure 3.9 shows corresponding examples. The examples show how sedans are most frequently matched to sedans (Figure 3.9(a)), racing bikes to racing bikes (Figure 3.9(b), top), and mountain bikes to mountain bikes (Figure 3.9(b), bottom).

3.6 Conclusions

We have designed a detailed 3D geometric object class model for 3D object recognition and modeling, complementing ideas from the early days of computer vision with modern techniques for robust model-to-image matching. Combining 3D wireframes with discriminative local shape detectors, we have demonstrated the successful recovery of detailed 3D object shape and pose from single input images. We believe that this high level of geometric detail in an important ingredient to advance scene-level reasoning beyond what can be achieved with box-level object class representations.

In an extensive experimental study on the object classes *car* and *bicycle*, we have quantified the ability of our proposed system to recover detailed geometric object hypotheses from single images. The model has been tested in four different settings, ranging from accurate 2D localization of object parts, through continuous pose estimation, to ultra-wide baseline matching and fine-grained categorization of car and bicycle types. Throughout, the performance is on par with or higher than previously reported results.

In the future, we plan to extend the present work in two directions, namely to explicitly handle occluded object parts, and to reason jointly over multiple instances of several object classes in the same scene, in order to exploit the additional constraints due to the

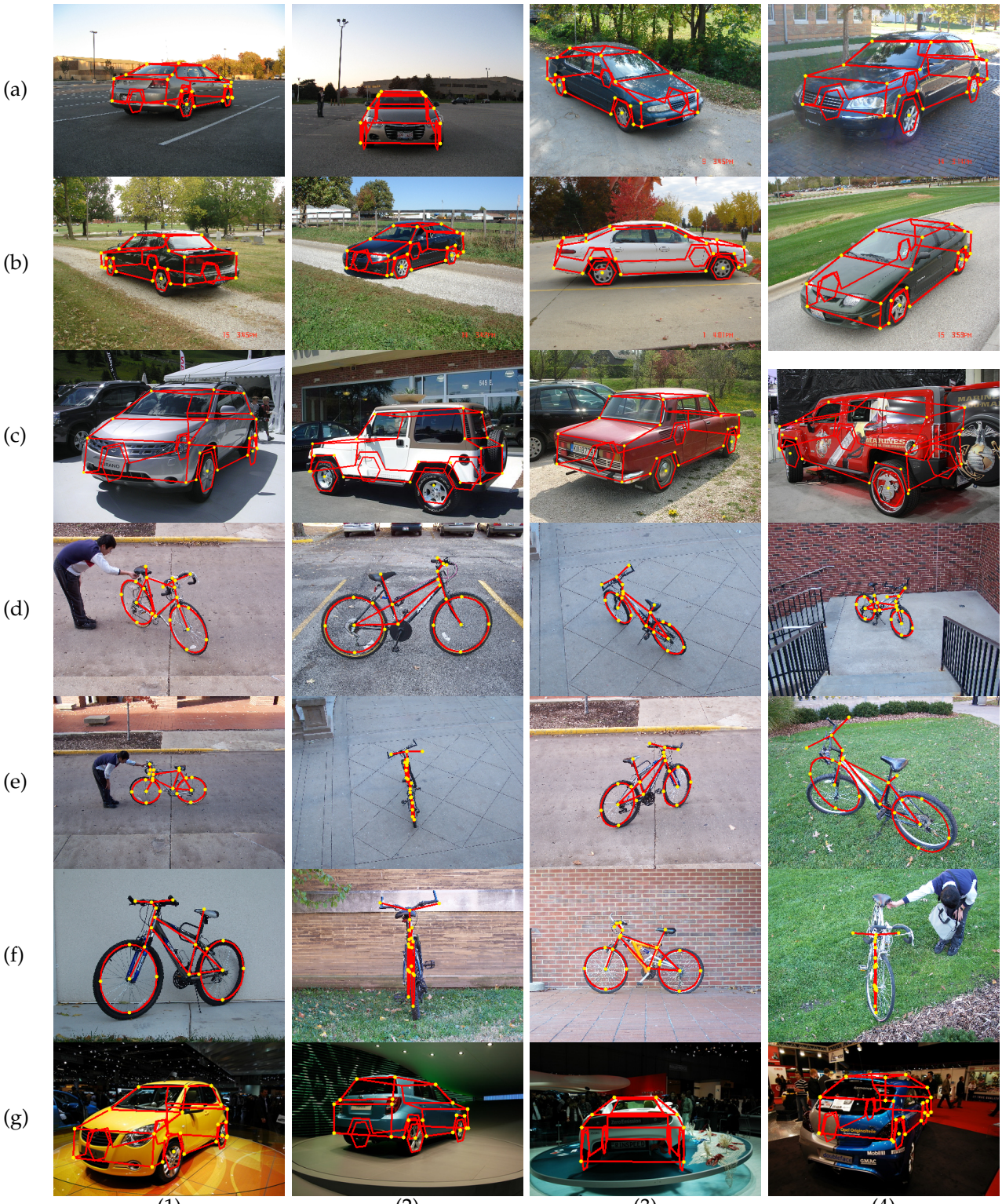


Figure 3.10: Example detections using the *full system*: estimated wireframes (yellow dots mark visible parts). *3D Object Classes* cars (rows (a), (b)), *Pascal VOC 2006* cars (row (c)), *3D Object Classes* bicycles (rows (d) - (f)), *EPFL Multi-view* cars (rows (g)). Successful detections (columns (1) - (3)), typical failure cases (column (4)).

Figure 3.11: Fully automatic 3D geometry estimation from single still images of *3D Object classes* cars (rows (a) - (c)) and bicycles ((d) - (f)) using the *full system* (edges of nearest database CAD models rendered in red; ground plane inferred from wheel positions). Ultra-wide baseline matching from car image pairs (row (g)).

common viewpoint as well as interactions between objects.

Acknowledgements We thank Bojan Pepik for providing his detections (Pepik et al.,

2012b) for use as initialization. This work has been supported by the Max Planck Center for Visual Computing and Communication.

Chapter 4

Explicit Occlusion Modeling for 3D Object Class Representations

M. Zeeshan Zia, Michael Stark, Konrad Schindler
26th IEEE Conference on Computer Vision and Pattern Recognition, 2013

(Author version; for typeset version please refer to the original conference paper.)

4.1 Abstract

Despite the success of current state-of-the-art object class detectors, severe occlusion remains a major challenge. This is particularly true for more geometrically expressive 3D object class representations. While these representations have attracted renewed interest for precise object pose estimation, the focus has mostly been on rather clean datasets, where occlusion is not an issue. In this paper, we tackle the challenge of modeling occlusion in the context of a 3D geometric object class model that is capable of fine-grained, part-level 3D object reconstruction. Following the intuition that 3D modeling should facilitate occlusion reasoning, we design an explicit representation of likely geometric occlusion patterns. Robustness is achieved by pooling image evidence from of a set of fixed part detectors as well as a non-parametric representation of part configurations in the spirit of *poselets*. We confirm the potential of our method on cars in a newly collected data set of inner-city street scenes with varying levels of occlusion, and demonstrate superior performance in occlusion estimation and part localization, compared to baselines that are unaware of occlusions.

4.2 Introduction

In recent years there has been a renewed interest in 3D object (class) models for recognition and detection. This trend has lead to a fruitful confluence of ideas from object detection on one side and 3D computer vision on the other side. State-of-the-art methods are not only capable of view-point invariant object categorization, but also give an estimate of the object's 3D pose (Savarese and Fei-Fei, 2007; Liebelt et al., 2008), and the locations of its parts (Li et al., 2011; Pepik et al., 2012a). Some go as far as

Figure 4.1: Fully automatic 3D shape, pose, and occlusion estimation.

estimating 3D wireframe models and continuous pose from single images (Zia et al., 2011; Leotta and Mundy, 2011; Zia et al., 2013).

Still, viewpoint-invariant detection and modeling is far from being solved, and several open research questions remain. Here, we focus on the problem of (partial) occlusion by other scene parts. Knowing the detailed part-level occlusion pattern of an object is valuable information both for the object detector itself and for higher-level scene models that use the object class model. In fact, 3D object detection under severe occlusions is still a largely open problem. Most detectors (Dalal and Triggs, 2005; Felzenszwalb et al., 2010) break down at occlusion levels of $\approx 20\%$.

However, when working with an explicit 3D representation of an object class, it should in principle be possible to estimate that pattern. Addressing self-occlusion is rather straight-forward with a 3D representation (Xiang and Savarese, 2012; Zia et al., 2011), since it is fully determined by the object shape and pose. On the other hand, inter-object occlusion is much harder to model, because it introduces relatively many additional unknowns (the occlusion states of all individual regions/parts of the object). Some part-based models resort to a data-driven strategy: every individual part can be occluded or unoccluded, and that latent state is estimated together with the object shape and pose (Li et al., 2011; Girshick et al., 2011).

Such a model has two weaknesses: first, it does not make any assumptions about the nature of the occluder, and can therefore lead to rather unlikely occlusion patterns (*e.g.* arbitrarily scattered small occluders). And second, it will have limited robustness, require careful tuning, and be hard to adapt to different scenarios. The latter is due to the tendency to simply label any individual part as occluded whenever it does not fit the evidence, and the associated brittle trade-off between the likelihood of occlusion and the uncertainty of the image evidence.

We argue that in many scenarios a per-part occlusion model is unnecessarily general. Rather, one can put a strong prior on the co-occurrence of part occlusions, because most

occluders are compact objects, and all one needs to know about them is the (also compact) projection of their outline onto the image plane. We therefore propose to restrict the possible occluders to a small finite set that can be explicitly enumerated, and to estimate the type of occluder and its location during inference. The very simple, but powerful intuition behind this is that *when restricted to compact regions inside the object's bounding box, the number of possible occlusion patterns is in fact very small.* Still such an occluder model is more general than one that only truncates the bounding box from left, right, above or below *e.g.* Wang et al. (2009); Enzweiler et al. (2010) or at image boundaries (Vedaldi and Zisserman, 2009), *cf.* Figure 4.2. *E.g.* , the proposed model can represent a vertical pole occluding the middle of the object, a frequent case in urban scenarios.

The contribution described in this paper is a viewpoint-invariant method for detailed reconstruction of severely occluded objects in monocular images. To obtain a complete framework for detection and reconstruction, the novel method is initialized with a variant of the *poselets* framework (Bourdev and Malik, 2009) adapted to the needs of our 3D object model. The object representation itself has three parts: a deformable shape model in the form of an active shape model defined over local object parts, an appearance model which integrates evidence from detectors for the parts as well as their configurations, and an occlusion model in the form of a set of occlusion masks. Experiments on images with strong occlusions show that the model can correctly infer even large occluders, and enables monocular 3D modeling in situations where representations without occlusion model fail.

4.3 Related work

In the early days of computer vision, 3D object models with a lot of geometric detail (Roberts, 1963; Brooks, 1981; Lowe, 1987; Sullivan et al., 1995) commanded a lot of interest, but unfortunately failed to tackle challenging real world imagery. Most current object class detectors provide coarse outputs in the form of 2D or 3D bounding boxes along with classification into a discrete set of viewpoints (Yan et al., 2007; Savarese and Fei-Fei, 2007; Liebelt et al., 2008; Felzenszwalb et al., 2010; Ozuysal et al., 2009; Stark et al., 2010; Villamizar et al., 2011; Payet and Todorovic, 2011; Glasner et al., 2011).

Recently, there has been renewed interest in providing geometrically more detailed outputs, with different degrees of geometric consistency across viewpoints (Li et al., 2011; Zia et al., 2011; Xiang and Savarese, 2012; Pepik et al., 2012a; Hejrati and Ramanan, 2012; Zia et al., 2013). Such models have the potential to enhance high-level reasoning about objects and scenes, *e.g.* Hoiem et al. (2008); Ess et al. (2009); Wang et al. (2010); Hedau et al. (2010); Wojek et al. (2010).

Unfortunately occlusion, which is one of the most challenging impediments to visual object class modeling, has largely remained untouched in the context of such fine-grained object models. Recent attempts at occlusion reasoning in 2D object recognition include modeling the visibility/occluder mask (Fransens et al., 2006; Wang et al., 2009; Vedaldi and Zisserman, 2009; Gao et al., 2011; Kwak et al., 2011), training detectors for occluded objects in specific frequently found configurations (Tang et al., 2012), using depth and/or

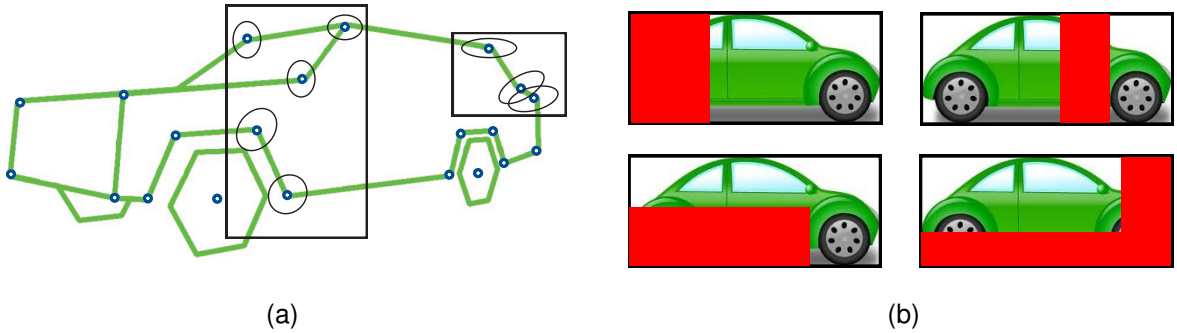
motion cues (Enzweiler et al., 2010; Meger et al., 2011), asserting an "occluder part" when part evidence is missing (Girshick et al., 2011), applying RANSAC to choose a subset of parts (Li et al., 2011), encoding occlusion states using local mixtures (Hejrati and Ramanan, 2012), and using a large number of partial object detectors which cluster together to give the full object (Bourdev and Malik, 2009), without explicit occluder modeling.

Fixed global object models have been known to give good results for fully visible object recognition (Dalal and Triggs, 2005), often outperforming part-based models. However, part-based models have unsurprisingly been found preferable for occlusion invariant detection (Bourdev and Malik, 2009; Girshick et al., 2011); in fact even when "global" models are extended to cope with occlusions (Wang et al., 2009; Kwak et al., 2011) they are divided into many local cells, which are effectively treated as parts with fixed relative locations. Part-based 3D object models with strong geometric constraints as Li et al. (2011); Zia et al. (2011) are thus strong candidates for part-level occlusion reasoning: they can cope with locally missing evidence, but still ensure the relative part placement always corresponds to a plausible global shape. On the downside, these are computationally fairly expensive models, therefore their evaluation on images in Li et al. (2011) is limited to a small bounding box around the object of interest. We thus propose a two-layer model, where objects are first detected with a variant of the *poselet* method (Bourdev and Malik, 2009) to obtain a rough localization and pose; then a detailed shape, pose and occlusion mask are inferred with an explicit 3D model as in Zia et al. (2011, 2013), which also includes the additional clues for part placement afforded by the preceding detector. Note that the two layers go together well, since spatially compact occluders will leave configurations of adjacent object parts ("poselets") visible.

4.4 Model

We propose to split 3D object detection and modeling into two layers. The first layer is a representation in the spirit of the *poselet* framework (Bourdev and Malik, 2009), *i.e.* a collection of viewpoint-dependent part $configurations$ tied together by relatively loose geometric constraints. The purpose of this layer is to find, in a large image, approximate 2D bounding boxes with rough initial estimates of the objects' pose. The part-based structure enables the model to deal with partial occlusion, and provides evidence for visible $configurations$ that can be used in the second layer.

The second layer is a 3D active shape model based on local $parts$, augmented with a collection of explicit occlusion masks. The ASM tightly constrains the object geometry to plausible shapes, and thus can more robustly predict object shape when parts are occluded, respectively predict the locations of the occluded parts. The model also includes the activations of the $configurations$ from the first layer as additional evidence, tying the two layers together.



(a) (b)

Figure 4.2: (a) Two larger part $configurations$ comprising of multiple smaller $parts$, as well as their relative distributions, (b) a few example occlusion masks.

Parts and part configurations

We start the explanation with the local appearance model. The atomic units of our representation are $parts$, which are small square patches located at salient points of the object. The patches are encoded with densely sampled shape-context descriptors (Andriluka et al., 2009), and a multi-class Random Forest is trained to recognize them. The classifier is viewpoint-invariant, meaning that one class label includes views of a part over all poses in which the part is visible (Zia et al., 2013). This marginalization over viewpoints speeds up part detection (which is the bottleneck of the method) by an order of magnitude[1] compared to individual per-viewpoint classifiers (Andriluka et al., 2009; Stark et al., 2010; Zia et al., 2011), while we did not observe a perfomance drop at the system level in spite of visibly blurrier part likelihoods. Additionally, the classifier also has a background class, which will be used for normalization (*cf.* Section4.4). Like Stark et al. (2010); Zia et al. (2011, 2013) we exploit the fact that with modern descriptors the part classifier can be trained mostly on synthetic renderings of 3D CAD models rather than on real data, which massively reduces the annotation effort.

The basic unit of the first layer are larger part $configurations$ ranging in size from 25% to 60% of the full object extent. These are defined in the spirit of *poselets*: Small sets of the local $parts$ described above are chosen and clustered (with standard k-means) according to the parts' spatial layout. The advantage of this clustering is that it discovers when object portions have high variability in appearance, *e.g.* the rear portion of sedans *vs.* hatchbacks as seen in a side view. To account for the spatial variability *within* a $configuration$, a single component DPM detector (Felzenszwalb et al., 2010) is trained for each configuration. We found that for these detectors real training data is needed, thus they are trained on annotated training images.

Geometric model

As explained earlier, we employ different geometric models for the initial detection and the subsequent 3D modeling. The first layer follows the philosophy of the ISM/poselet method. For each configuration the mean offset from the object centroid as well as the

[1] Also, training is two orders of magnitude faster.

mean relative scale are stored during training, and at test time detected $configurations$ cast a vote for the object center and scale. These votes are then combined via greedy agglomerative clustering, similar to Bourdev and Malik (2009). After non-maximum suppression, the output of the first layer consists of a set of approximate 2D bounding boxes, each with a coarse pose estimate (quantized to 8 canonical viewpoints) and a list of activated $configurations$.

The second layer utilizes a more explicit representation of global object geometry that is better suited for estimating detailed 3D object shape and pose. In the tradition of *active shape models* we learn a deformable 3D wireframe from annotated 3D CAD models, like in Zia et al. (2011, 2013). The wireframe model is defined through an ordered collection of n vertices in 3D-space, chosen at salient points on the object surface in a fixed topological layout. Following standard point-based shape analysis (Cootes et al., 1995) the object shape and variability are represented as the sum of a mean wireframe μ and deformations along r principal component directions $\mathbf{p}_j$. The geometry parameters s_k determine the amount of deviation from the mean shape (in units of standard deviation σ_j along the respective directions): $\mathbf{X}(\mathbf{s}) = \mu + \sum_{k=1}^{r} s_k \sigma_k \mathbf{p}_k + \epsilon$. The $parts$ described above are defined as small windows around the 2D projection of such a vertex ($\approx$ 10% in size of the full object width). The parts cover the full extent of the represented object class, thus they allow for fine-grained estimation of 3D geometry and continuous pose, as well as for detailed reasoning about occlusion relations. We point out once more that these parts are viewpoint-independent, *i.e.* a part covers the appearance of a vertex over the entire viewing sphere.

Explicit occluder representation

While the first layer contains only implicit information about occluders (in the form of supposedly visible, but undetected $configurations$), the second layer includes an explicit occluder representation. Occluders are assumed to block the view onto a spatially connected region of the object. Due to the object being modeled as a sparse collection of parts, occluders can only be distinguished if the visibility of at least one part changes, which further reduces the space of possible occluders. Thus, one can well approximate the set of all occluders by a discrete set of occlusion masks a (for convenience we denote the empty mask which leaves the object fully visible by a_0). Figure 4.2(b) shows exemplary occlusion masks.

With that set, we aim to explicitly recover the occlusion pattern during second-layer inference, by selecting one of the masks. All parts falling inside the occlusion mask are considered occluded, and consequently their detection scores are not considered in the objective function (Section 4.4). Instead, they are assigned a fixed low score, corresponding to a weak uniform prior that prefers parts to be visible and counters the bias to “hide behind the occluder”.

Occlusion of parts is modeled by indicator functions $o_j(\mathbf{s}, \boldsymbol{\theta}, a)$, where j represents the part index, $\mathbf{s}$ represents the object geometry (Section 4.4) and $\boldsymbol{\theta}$ the viewpoint. The set of masks a_i act as a prior that specifies which parts occlusions can co-occur. For completeness we mention that object self-occlusion is modeled with the same indicator variables,

but does not require separate treatment, since it is completely determined by shape and pose.

Shape, pose, and occlusion estimation

During inference, we attempt to find instances of the 3D shape model and of the occlusion mask that best explain the observed image evidence. Recall that we wish to estimate an object's 3D pose (5 parameters, assuming no in-plane rotation), geometric shape (7 ASM shape parameters), and an occluder index (1 parameter). Taken together, we are faced with a 13-dimensional search problem, which would be prohibitively expensive even for a moderate image size. We therefore first cut down the search space in the first layer with a simpler and more robust object detection step, and then fit the full model locally at a small number of (candidate) detections.

First layer inference starts by detecting instances of our part $configurations$ in the image with the corresponding DPM detectors. Each detected configuration casts an associated vote for the full object 2D location and scale $\mathbf{q} = (q_x, q_y, q_s)$, and for the pose $\boldsymbol{\theta} = (\theta_{az}, \theta_{el})$. At this point, the azimuth angle is restricted to a small set of discrete steps and the elevation angle is fixed, both to be refined in the second layer. The votes are clustered with a greedy agglomerative clustering scheme as in Bourdev and Malik (2009) to obtain detection hypotheses $\mathcal{H}$, each with a list of contributing configurations $\{l_1 \dots l_p\}$ that voted for the object's presence.

Part location prediction from first layer. Since the configurations are made up of multiple parts confined to a specific layout with little spatial variability (Section 4.4), their detected instances l_i already provide some information about the part locations in image space. The means $\boldsymbol{\mu}_{ij}$ and covariances $\boldsymbol{\sigma}^2_{ij}$ of the parts' locations within a configuration's bounding box are estimated from the training data, and v_{ij} are binary flags indicating which parts j are found within the $configuration\ l_i$. Figure 4.2(a) illustrates two such larger $configurations$, whose detection can be used to predict the location of the constituent parts as gaussian distributions with the respective means and covariances relative to the bounding box of the configuration.[2]

Second layer objective function. After evaluating the first layer of the model we are left with a sparse set of (putative) detections, such that we can afford to evaluate a relatively expensive objective function. We denote an object instance by $\mathbf{h} = (\mathbf{s}, f, \boldsymbol{\theta}, \mathbf{q}, a)$, comprising of shape parameters $\mathbf{s}$ (eqn. 4.4), camera focal length f, viewpoint parameters for azimuth and elevation $\boldsymbol{\theta}$, and translation and scale parameters in image space $\mathbf{q}$. The projection matrix P that maps the 3D vertices $\mathbf{X}_j(\mathbf{s})$ to image points $\mathbf{x}_j$ is assumed to depend only on $\boldsymbol{\theta}$, and $\mathbf{q}$, while f is fixed, assuming similar perspective effects for all images: $\mathbf{x}_j = \mathrm{P}(f, \boldsymbol{\theta}, \mathbf{q})\mathbf{X}_j(\mathbf{s})$.

[2] In practice it is beneficial to only use configurations whose part predicitons are sufficiently accurate, as determined by cross-validation.

Fitting the model amounts to finding a MAP-estimate of the objective function $\mathcal{L}(\mathbf{h})$:

$$\hat{\mathbf{h}} = \arg\max_h \left[\mathcal{L}(\mathbf{h})\right] \ , \tag{4.1}$$

$$\mathcal{L}(\mathbf{h}) = \max_{\boldsymbol{\varsigma}} \left[\frac{1}{\sum_{j=1}^{m} o_j(\mathbf{s}, \boldsymbol{\theta}, a_0)} \sum_{j=1}^{m} \Big(\mathcal{L}_v + \mathcal{L}_o + \mathcal{L}_c \Big) \right]. \tag{4.2}$$

The factor $1/\sum_{j=1}^{m} o_j(\mathbf{s}, \boldsymbol{\theta}, a_0)$ normalizes for the varying number of self-occluded parts at different viewpoints. For each potentially visible part there are three terms: $\mathcal{L}_v$ is the evidence $S_j(\boldsymbol{\varsigma}, \mathbf{x}_j)$ for part j if it is visible, found by looking up the detection score at image location $\mathbf{x}_j$ and scale $\boldsymbol{\varsigma}$. Part likelihoods are normalized with the background score $S_b(\boldsymbol{\varsigma}, \mathbf{x}_j)$, as in Villamizar et al. (2011). $\mathcal{L}_o$ assigns a fixed likelihood c to the part, if it lies under the occlusion mask. $\mathcal{L}_c$ measures how well the part j is predicted by the larger $configurations$.

$$\mathcal{L}_v = o_j(\mathbf{s}, \boldsymbol{\theta}, a) \log \frac{S_j(\boldsymbol{\varsigma}, \mathbf{x}_j)}{S_b(\boldsymbol{\varsigma}, \mathbf{x}_j)} \ , \tag{4.3}$$

$$\mathcal{L}_o = \big(o_j(\mathbf{s}, \boldsymbol{\theta}, a_0) - o_j(\mathbf{s}, \boldsymbol{\theta}, a) \big) c \ , \tag{4.4}$$

$$\mathcal{L}_c = \frac{o_j(\mathbf{s}, \boldsymbol{\theta}, a)}{p} \sum_{i=1}^{p} v_{ij} \log \left(1 + \lambda \mathcal{N}(\mathbf{x}_j; \boldsymbol{\mu}_{ij}, \boldsymbol{\sigma}_{ij}^2) \right) \ . \tag{4.5}$$

Second layer inference. The objective (4.2) is a mixed discrete-continuous function which is neither convex nor smooth, and thus cannot be easily maximized. We find an approximate MAP-estimate $\hat{\mathbf{h}}$ with sample-based stochastic hill-climbing. Specifically, we maintain a set of weighted samples (particles), each corresponding to a distinct set of values in the space of object hypotheses $\{\mathbf{s}, \boldsymbol{\theta}, \mathbf{q}, a\}$. Particles are iteratively updated, by re-sampling individual parameters from independent Gaussians centered at the current values, similar to Leordeanu and Hebert (2008). In our scheme the variances of these Gaussians are gradually reduced according to a fixed annealing schedule. Other than the remaining parameters, the mask indices a are discrete and have no obvious ordering. To define similarity between them we sort the set of masks w.r.t. the Hamming distance from the current one, then we sample the offset in this ordering from a Gaussian.

The inference is initialized at the location, scale and pose returned by the first layer, while the initial shape parameters are chosen randomly and the occlusion mask is set to a_0.

4.5 Experiments

In the following, we evaluate the performance of our approach in detail, focusing on its ability to recover fine-grained, part-level accurate object shape and accompanying occlusion estimates. In particular, we quantify the ability of our method to localize entire objects (Section 4.5), to localize their constituent parts (Section 4.5), and to estimate occluded object portions (in the form of part occlusion labels), for varying levels of occlusion (Section 4.5).

The free parameters for (4.4) and (4.5) are estimated by cross-validation on the 3D Object Classes (Savarese and Fei-Fei, 2007), for which part level annotations are publicly available (Zia et al., 2011). The set of 288 occlusion masks has been generated automatically and pruned manually to exclude very unlikely masks.

Data set

As a testbed we have collected a novel, challenging data set of inner-city street scenes. It consists of 101 images of resolution 2 mega-pixels, showing street scenes with cars, with occlusions ranging from 0% to >60% of the bounding box as well as the parts. Although there are several publicly available car datasets, none of them is suitable for our purposes, since we found that part detector performance deteriorates significantly for objects smaller than 60 pixels in height. Some datasets do not contain occluded cars (*e.g.* 3D Object Classes (Savarese and Fei-Fei, 2007), EPFL Multiview Cars (Ozuysal et al., 2009)); others do, but have rather low resolution (Ford Campus Vision and Lidar, Pascal VOC (Everingham et al., 2010)), which makes them unsuitable for detailed geometric model fitting – and also seems unrealistic, given today's omnipresent high-resolution cameras. We further opted for taking the pictures ourselves, in order to avoid the strong bias of internet search towards high-contrast, high-saturation images. Figures 4.4,4.5 show example images from the new data set.[3]

Model variants and baselines

We evaluate and compare the performance of the following competing models: *(i)* a naive baseline without 3D estimation, which places a fixed canonical 3D car (the mean of our active shape model) inside the detected first-layer bounding box in the estimated (discrete) pose. *(ii)* the ASM model of Chapter 3, which corresponds to the second layer of our model without any form of occlusion reasoning (*i.e.* assuming that all parts are visible except for self-occlusions), and without using the part $configurations$ from the first layer. *(iii)* the proposed model, including prediction of occluders, but not using the $configurations$ during second-layer inference. *(iv)* our full model with occluder prediction and leveraging additional evidence from $configurations$ for second-layer inference.

Object localization

We commence by verifying that our first layer, *i.e.* a combination of DPM *configuration* detectors and *poselet*-style voting, is competitive with alternative algorithms for detecting objects in 2D. To that end we compare our first layer, trained on a dataset comprising of around 1000 full car images downloaded from the internet, with the original poselet implementation (Bourdev and Malik, 2009) pre-trained on Pascal VOC (Everingham et al., 2010) (training code for Bourdev and Malik (2009) is not publicly available). We also include the deformable part model (DPM, Felzenszwalb et al., 2010), both trained on the same 1000 car images (using default parameters), as well as the pre-trained model (on Pascal VOC (Everingham et al., 2010)), as a popular state-of-the-art reference. Unfortunately neither of these implementations directly outputs a viewpoint.

[3]The data set, along with all training data and annotations, pre-trained models, and source code is made available at http://www.igp.ethz.ch/photogrammetry/downloads.

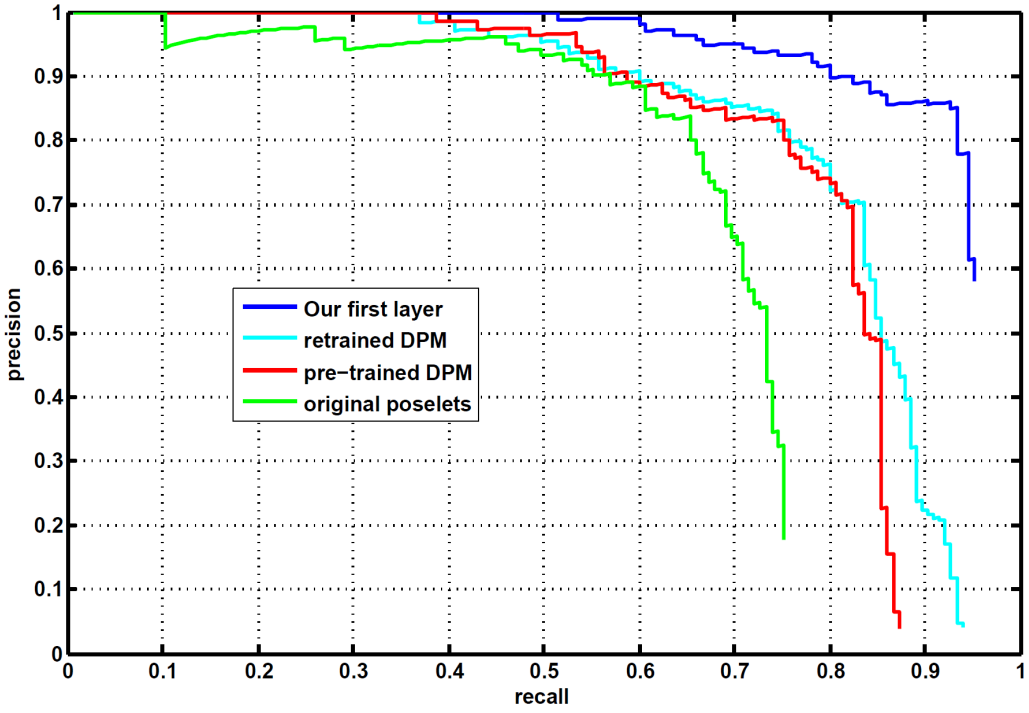


Figure 4.3: Object detection acuracy of different 2D detectors.

Protocol. We follow the classical object detection protocol of Pascal VOC (Everingham et al., 2010), plotting precision *vs.* recall for 50% intersection-over-union between predicted and ground truth bounding box.

Results. Precision-recall curves are shown in Figure 4.3. We observe that the original poselets (Bourdev and Malik, 2009) already perform reasonably well on our data (67% AP). The pre-trained DPM (Felzenszwalb et al., 2010) improves the results to 76% AP , and the retrained model, to 79% AP. Our first layer outperforms both by a significant margin, achieving 88% AP, which we consider a solid basis for the subsequent 3D inference. In particular we point out that the combination of a strong part detector with Hough-style voting reaches high recall (up to 95%) at reasonable precision. The fact that only few instances are irrevocably lost in the first layer confirms that splitting into a coarse detection layer and a detailed modeling layer is a viable approach (see Table 4.1).

	Full dataset	$<80\%$ visibility	$<60\%$ visibility
Total cars	165	96	48
Detected	147	85	42

Table 4.1: First-layer detection results (bounding box and 1D pose). Subsequent second-layer results are given for the detected instances (line “detected”).

Occlusion estimation

We proceed by evaluating how well our model can distinguish between occluded and unoccluded parts. Note that while this ability is potentially also useful for further reasoning about the occluder, its primary importance here lies in the 3D object modeling itself: a good estimate of the part-level occlusion state is necessary in order not to mistakenly use evidence from background structures, and hence forms the basis for recovering the objects' 3D extent and shape.

Protocol. The predicted part occlusions are evaluated as two-class classification: we first remove all self-occluded $parts$, and then compare occlusion labeling o_j induced by the estimated occluder a_i to ground truth annotations.

Results. Table 4.2 shows the percentages of correctly inferred part occlusions. First, we observe that the acuracy decreases with increasing occlusion level, matching our intuition. Baseline 1 is obviously not applicable, since it offers no possibility to decide about part-level occlusion. To make the second baseline comparable, which also does not make occlusions explicit, we place a threshold (equal in value to c used in the likelihood (4.2)) on part detection scores and call parts with too low scores occluded. Although that heuristic works surprisingly well, our occlusion inference outperforms the baseline by significant margins (4.5 – 5.9%) for all levels of occlusion. Additionally using the active $configurations$ from the first layer during inference boosts classification performance by a further 1.2 – 3.0%. We point out that the additional evidence provided by the larger $configurations$ is most beneficial at high levels of occlusion, and that even for heavily occluded vehicles that are only 30 – 60% visible, 83.1% of the part occlusions are correctly predicted.

	Full dataset	$<$ 80% visibility	$<$ 60 % visibility
baseline 1	—	—	—
baseline 2 (Zia et al., 2011, 2013)	79.5%	76.7%	75.6%
w/o configurations (ours)	84.4%	82.6%	80.1%
w/ configurations (ours)	**85.6%**	**84.7%**	**83.1%**

Table 4.2: Part-level occlusion prediction (percentage of correctly classified parts). See text for details.

Part localization

The primary goal of our occlusion model is better 3D object modeling: we wish to correctly predict objects' spatial extent, shape and pose, to support higher-level tasks such as monocular depth estimation, free-space computation and physically plausible, collision-free scene understanding. To quantify the ability to recover 3D extent and shape, we assess how well individual parts of the 3D geometric model can be localized. Since we have no 3D ground truth, part localization accuracy is measured in the 2D image plane by comparing to manual annotations.

Protocol. We follow the common evaluation protocol of human body pose estimation

and report the average percentage of correctly localized parts, using a relative threshold adjusted to the size of the car. The threshold is set to 20 pixels for a car of size 500×170 pixels, *i.e.* $\approx 4\%$ of the total length.

Results. Part localization results for different levels of oclusion are given in Table 4.3. We make the following observations. First, baseline 1 performs poorly, *i.e.* the bounding box and pose predictions of a 2D detector and/or a rigid average car are insufficient. Second our occlusion-aware approach outperforms the 3D-ASM of Zia et al. (2011, 2013) without occlusion modeling by 2.5% on the entire dataset, and that margin increases to 5.3% for the heavily occluded cars. Third, adding evidence form $configurations$ brings only a small improvement for the full dataset, but the improvement is more pronounced for heavier occlusions. Finally, we manage to sucessfully localize $> 80\%$ of the parts even at occlusion levels of 40% or more.

Figure 4.5 shows qualitative examples, highlighting the differences between the naive baseline 1, the baseline approach without occlusion modeling (Zia et al., 2011, 2013), and the two evaluated variants of our model. Clearly, the fits without occlusion model are severely disturbed in the presence of even moderate occlusion. Our approach without $configurations$ seems to perform as well as the full model when it comes to predicting the occluder, but is slightly more prone to mistakes concerning the overall object shape (e.g., rows a, b). Figure 4.4 shows further qualitative results of the full model.

	Full dataset	$<$ 80% visibility	$<$ 60 % visibility
baseline 1	32.0%	33.6%	39.7%
baseline 2 (Zia et al., 2011, 2013)	80.0%	75.6%	74.5%
w/o configurations (ours)	82.5%	80.0%	79.8%
w/ configurations (ours)	**82.7%**	**80.7%**	**83.5%**

Table 4.3: Part localization accuracy (percentage of correctly localized parts). See text for details.

4.6 Conclusion

We have explored the problem of occlusion in the context of geometric, part-based 3D object class representations for object detection and modeling. We have proposed a two-layer model, consisting of a robust, but coarse 2D object detector, followed by a detailed 3D model of pose and shape. The first layer accumulates votes from view-point dependent part $configurations$, such that it can tolerate quite large degrees of occlusion, but does not explicitly detect them. The second layer combines an explicit deformable 3D shape model over smaller $parts$ with evidence from the first-level $configurations$, as well as with an explicit occlusion model in the form of a collection of possible occlusion masks. Although that representation of occlusion is rather simple, experiments on detecting and modeling cars in a dataset of street scenes have confirmed the model to correctly

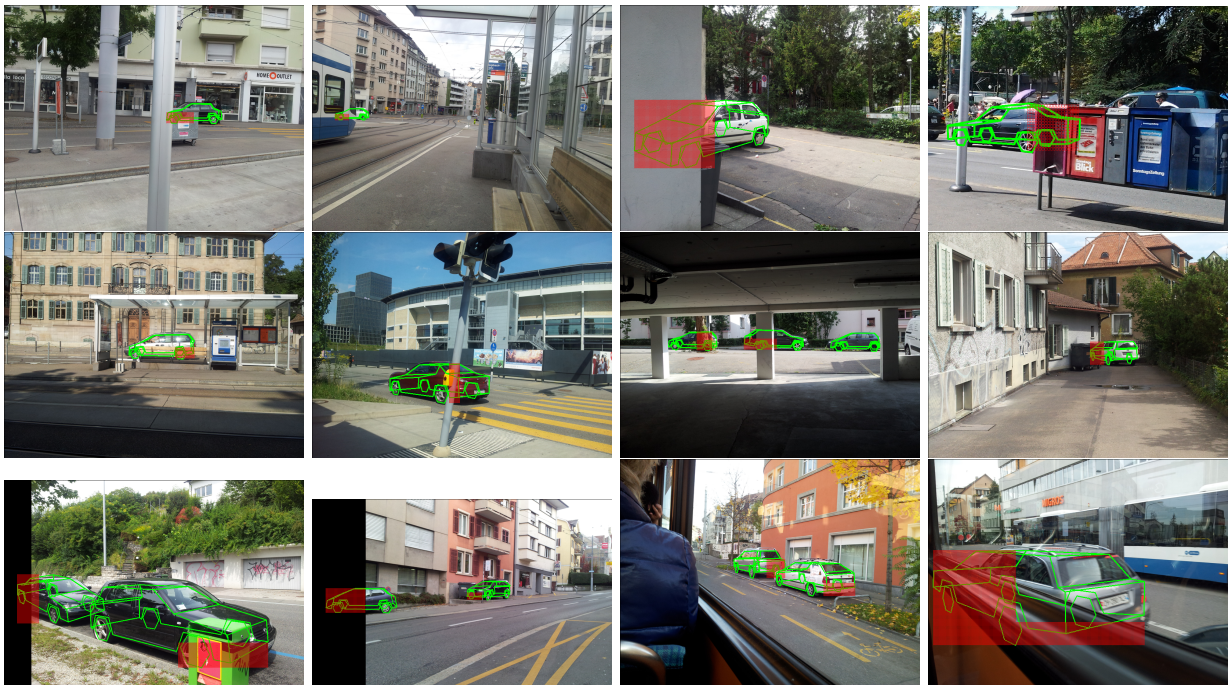


Figure 4.4: Example detections using our full system.

estimate both the occlusion pattern and the car shape and pose even under severe occlusion, clearly outperforming a baseline that is agnostic about occlusions.

Acknowledgements. This work has been supported by the Max Planck Center for Visual Computing and Communication.

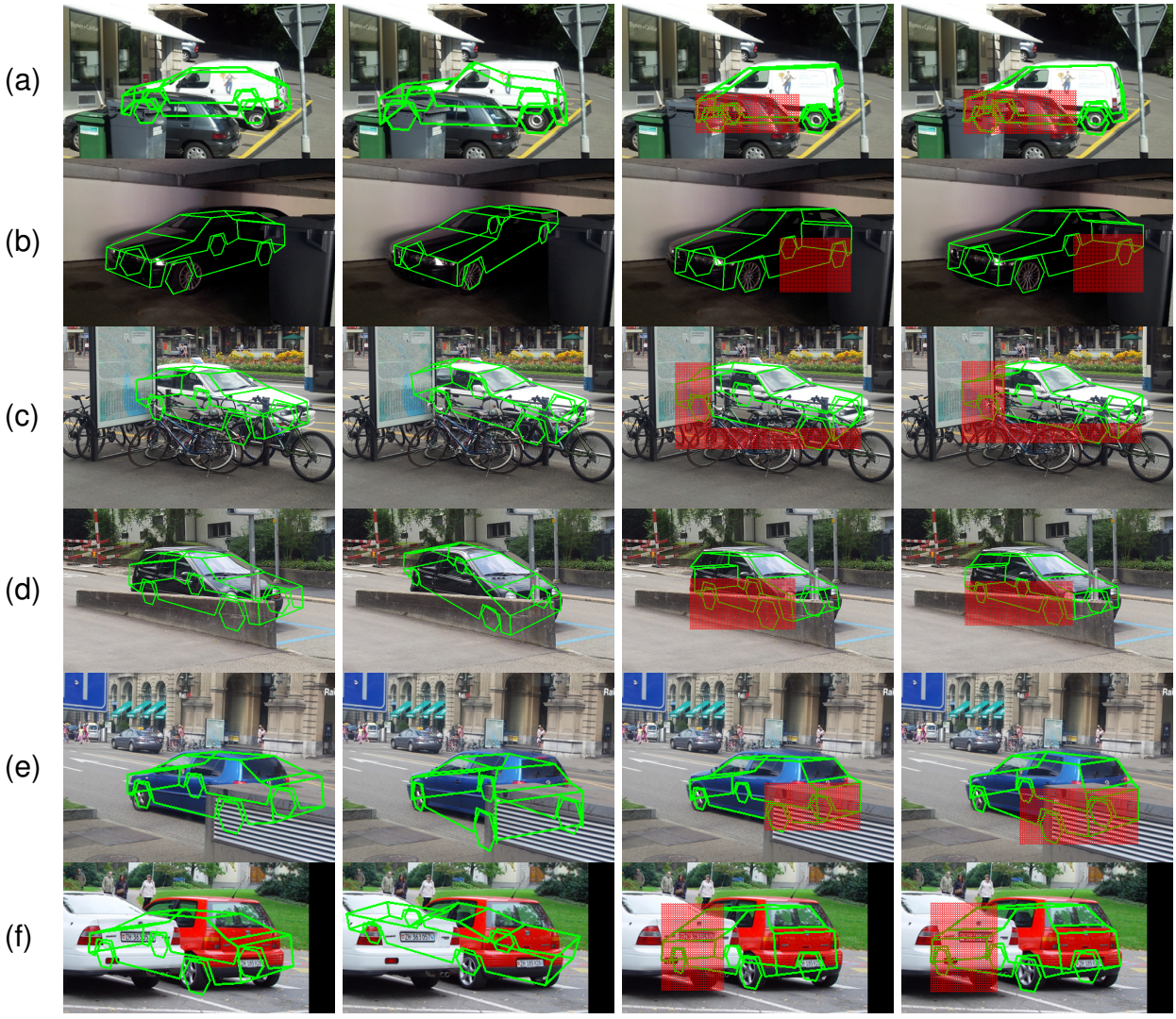


Figure 4.5: Comparing model fits: canonical car shape in detected bounding box *i.e.* baseline 1 (column 1), baseline 2 (Zia et al., 2011, 2013) (column 2), without *poselets* (column 3), with *poselets* (column 4).

Chapter 5

Towards Scene Understanding with Detailed 3D Object Representations

M. Zeeshan Zia, Michael Stark, Konrad Schindler
Submitted to journal

(Author version; for typeset version please refer to the original journal paper.)

5.1 Abstract

Current approaches to semantic image and scene understanding typically employ rather simple object representations such as 2D or 3D bounding boxes. While such coarse models are robust and allow for reliable object detection, they discard much of the information about objects' 3D shape and pose, and thus do not lend themselves well to higher-level reasoning. Here, we propose to base scene understanding on a high-resolution object representation. An object class – in our case cars — is modeled as a deformable 3D wireframe, which enables fine-grained modeling at the level of individual vertices and faces. We augment that model to explicitly include vertex-level occlusion, and embed all instances in a common coordinate frame, in order to infer and exploit object-object interactions. Specifically, from a single view we jointly estimate the shapes and poses of multiple objects in a common 3D frame. A ground plane in that frame is estimated by consensus among different objects, which significantly stabilizes monocular 3D pose estimation. The fine-grained model, in conjunction with the explicit 3D scene model, further allows one to infer part-level occlusions between the modeled objects, as well as occlusions by other, unmodeled scene elements. To demonstrate the benefits of such detailed object class models in the context of scene understanding we systematically evaluate our approach on the challenging KITTI street scene dataset. The experiments show that the model's ability to utilize image evidence at the level of individual parts improves monocular 3D pose estimation w.r.t. both location and (continuous) viewpoint.

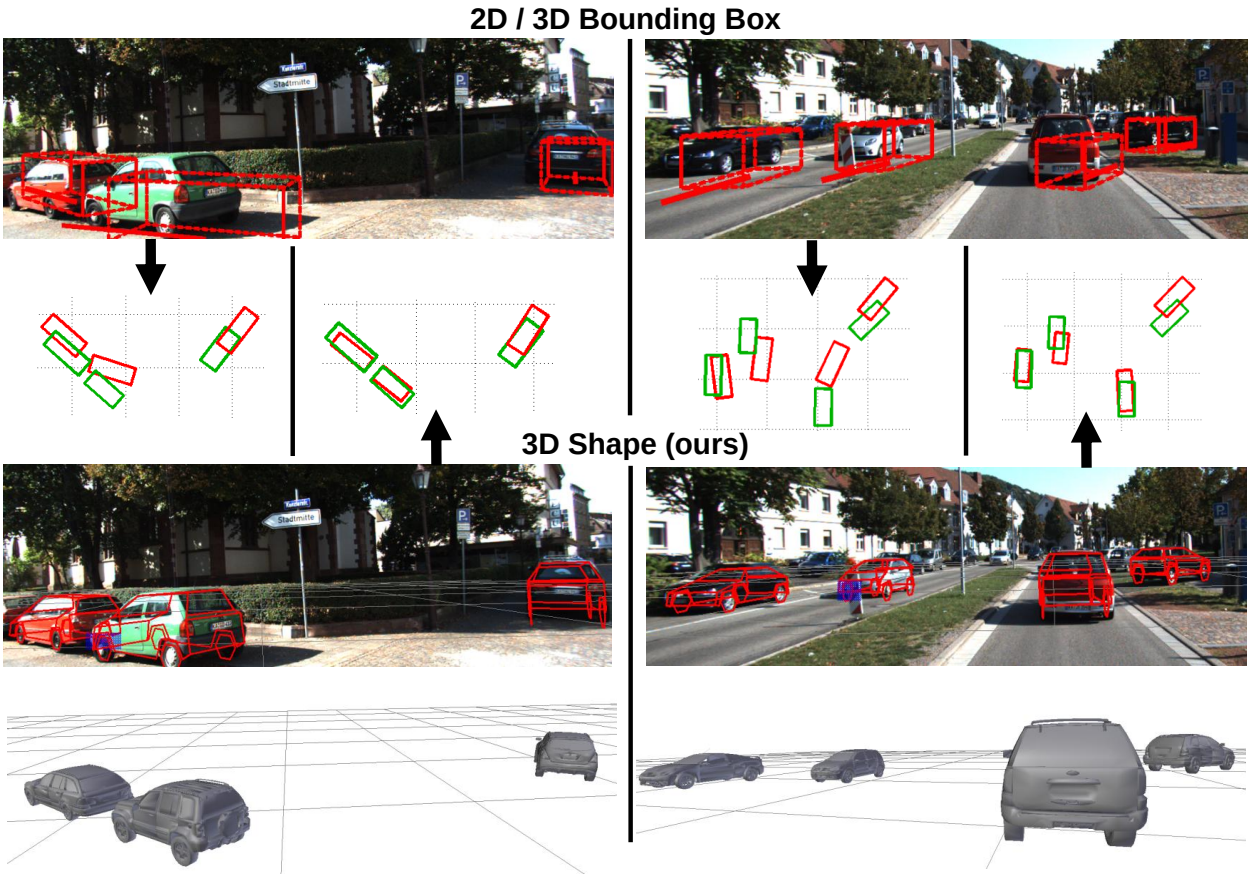


Figure 5.1: *Top:* Coarse 3D object bounding boxes derived from 2D bounding box detections (not shown). *Bottom:* our fine-grained 3D shape model fits improve 3D localization (see bird's eye views).

5.2 Introduction

The last ten years have witnessed great progress in automatic visual recognition and image understanding, driven by advances in local appearance descriptors, the adoption of discriminative classifiers, and more efficient techniques for probabilistic inference. In several different application domains we now have semantic vision sub-systems that work on real-world images. Such powerful tools have sparked a renewed interest in the grand challenge of visual 3D scene understanding. Meanwhile, individual object detection performance has reached a plateau after a decade of steady gains (Everingham et al., 2010), further emphasizing the need for contextual reasoning.

A number of geometrically rather coarse scene-level reasoning systems have been proposed over the past few years (Hoiem et al., 2008; Wang et al., 2010; Hedau et al., 2010; Gupta et al., 2010; Silberman et al., 2012), which apart from adding more holistic scene understanding also improve object recognition. The addition of context and the step to reasoning in 3D (albeit coarsely) makes it possible for different vision sub-systems to interact and improve each other's estimates, such that the sum is greater than the parts. Very recently, researchers have started to go one step further and increase the level-

of-detail of such integrated models, in order to make better use of the image evidence. Such models learn not only 2D object appearance but also detailed 3D shape (Xiang and Savarese, 2012; Hejrati and Ramanan, 2012; Zia et al., 2013). The added detail in the representation, typically in the form of wireframe meshes learned from 3D CAD models, makes it possible to also reason at higher resolution: beyond measuring image evidence at the level of individual vertices/parts one can also handle relations between parts, *e.g.* shape deformation and part-level occlusion (Zia et al., 2013). Initial results are encouraging. It appears that the more detailed scene interpretation can be obtained at a minimal penalty in terms of robustness (detection rate), so that researchers are beginning to employ richer object models to different scene understanding tasks (Choi et al., 2013; Del Pero et al., 2013; Zhao and Zhu, 2013; Xiang and Savarese, 2013; Zia et al., 2014a).

Here we describe one such novel system for scene understanding based on monocular images. Our focus lies on exploring the potential of jointly reasoning about multiple objects in a common 3D frame, and the benefits of part-level occlusion estimates afforded by the detailed representation. We have shown in previous work (Zia et al., 2013) how a detailed 3D object model enables a richer pseudo-3D $(x, y, scale)$ interpretation of simple scenes dominated by a single, unoccluded object—including fine-grained categorization, model-based segmentation, and monocular reconstruction of a ground plane. Here, we lift that system to true 3D, *i.e.* CAD models are scaled to their true dimensions in world units and placed in a common, metric 3D coordinate frame. This allows one to reason about geometric constraints between multiple objects as well as mutual occlusions, at the level of individual wireframe vertices.

Contributions. We make the following contributions.

First, we propose a viewpoint-invariant method for 3D reconstruction (shape and pose estimation) of severely occluded objects in single-view images. To obtain a complete framework for detection and reconstruction, the novel method is bootstrapped with a variant of the poselets framework (Bourdev and Malik, 2009) adapted to the needs of our 3D object model.

Second, we reconstruct scenes consisting of multiple such objects, each with their individual shape and pose, in a single inference framework, including geometric constraints between them in the form of a common ground plane. Notably, reconstructing the fine detail of each object also improves the 3D pose estimates (location as well as viewpoint) for entire objects over a 3D bounding box baseline (Figure 5.1).

Third, we leverage the rich detail of the 3D representation for occlusion reasoning at the individual vertex level, combining (deterministic) occlusion by other detected objects with a (probabilistic) generative model of further, unknown occluders. Again, integrated scene understanding yields improved 3D localization compared to independently estimating occlusions for each individual object.

And *fourth*, we present a systematic experimental study on the challenging KITTI street scene dataset (Geiger et al., 2012). While our fine-grained 3D scene representation can not yet compete with technically mature 2D bounding box detectors in terms of recall, it offers superior 3D pose estimation, correctly localizing $> 43\%$ of the detected cars up to 1 m and $> 55\%$ up to 1.5 m, even when they are heavily occluded.

Parts of this work appear in two preliminary conference papers (Zia et al., 2013, 2014a).

The present paper describes our approach in more detail, extends the experimental analysis, and describes the two contributions (extension of the basic model to occlusions, respectively scene constraints) in a unified manner.

The remainder of this paper is structured as follows. Section 3 reviews related work. Section 4 introduces our 3D geometric object class model, extended in Section 5 to entire scenes. Section 6 gives experimental results, and Section 7 concludes the paper.

5.3 Related work

Detailed 3D object representations. Since the early days of computer vision research, detailed and complex models of object geometry were developed to solve object recognition in general settings, taking into account viewpoint, occlusion, and intra-class variation. Notable examples include the works of Kanade (1980) and Malik (1987), who lift line drawings of 3D objects by classifying the lines and their intersections to common occurring configurations; and the classic works of Brooks (1981) and Pentland (1986), who represent complex objects by combinations of atomic shapes, generalized cones and super-quadrics. Matching CAD-like models to image edges also made it possible to address partially occluded objects (Lowe, 1987) and intra-class variation (Sullivan et al., 1995).

Unfortunately, such systems could not robustly handle real world imagery, and largely failed outside controlled lab environments. In the decade that followed researchers moved to simpler models, sacrificing geometric fidelity to robustify the matching of the models to image evidence—eventually reaching a point where the best-performing image understanding methods were on one hand bag-of-features models without any geometric layout, and on the other hand object templates without any flexibility (largely thanks to advances in local region descriptors and statistical learning).

However, over the past years researchers have gradually started to re-introduce more and more geometric structure in object class models and improve their performance (*e.g.* Leibe et al., 2006; Felzenszwalb et al., 2010). At present we witness a trend to take the idea even further and revive highly detailed deformable wireframe models (Zia et al., 2009; Li et al., 2011; Zia et al., 2013; Xiang and Savarese, 2012; Hejrati and Ramanan, 2012). In this line of work, object class models are learnt from either 3D CAD data (Zia et al., 2009, 2013) or images (Li et al., 2011). Alternatively, objects are represented as collections of planar segments (also learnt from CAD models, Xiang and Savarese, 2012) and lifted to 3D with non-rigid structure-from-motion. In this paper, we will demonstrate that such fine-grained modelling also better supports scene-level reasoning.

Occlusion modeling. While several authors have investigated the problem of occlusion in recent years, little work on occlusions exists for detailed part-based 3D models, notable exceptions being (Li et al., 2011; Hejrati and Ramanan, 2012).

Most efforts concentrate on 2D bounding box detectors in the spirit of HOG (Dalal and Triggs, 2005). Fransens et al. (2006) model occlusions with a binary visibility map over a

fixed object window and infer the map with expectation-maximization. In a similar fashion, sub-blocks that make up the window descriptor are sometimes classified into occluded and non-occluded ones (Wang et al., 2009; Gao et al., 2011; Kwak et al., 2011). Vedaldi and Zisserman (2009) use a structured output model to explicitly account for truncation at image borders and predict a truncation mask at both training and test time. If available, motion (Enzweiler et al., 2010) and/or depth (Meger et al., 2011) can serve as additional cues to determine occlusion, since discontinuities in the depth and motion fields are more reliable indicators of occlusion boundaries than texture edges.

Even though quite some effort has gone into occlusion invariance for global object templates, it is not surprising that part-based models have been found to be better suited for the task. In fact even fixed windows are typically divided into regular grid cells that one could regard as “parts" (Wang et al., 2009; Gao et al., 2011; Kwak et al., 2011). More flexible models include dedicated DPMs for commonly occuring object-object occlusion cases (Tang et al., 2012) and variants of the extended DPM formulation (Girshick et al., 2011), in which an occluder is inferred from the absence of part evidence. Another strategy is to learn a very large number of partial configurations (“poselets") through clustering (Bourdev and Malik, 2009), which will naturally also include frequent occlusion patterns. The most obvious manner to handle occlusion in a proper part-based model is to explicitly estimate the oclusion states of the individual parts, either via RANSAC-style sampling to find unoccluded ones (Li et al., 2011), or via local mixtures (Hejrati and Ramanan, 2012). Here we also store a binary occlusion flag per part, but explicitly enumerate allowable occlusion patterns and restrict the inference to that set.

Qualitative scene representations. Beyond detailed geometric models of individual objects, early computer vision research also attempted to model entire scenes in 3D with considerable detail. In fact the first PhD thesis in computer vision (Roberts, 1963) modeled scenes comprising of polyhedral objects, considering self-occlusions as well as combining multiple simple shapes to obtain complex objects. Koller et al. (1993) used simplified 3D models of multiple vehicles to track them in road scenes, whereas Haag and Nagel (1999) included scene elements such as trees and buildings, in the form of polyhedral models, to estimate their shadows falling on the road, as well as vehicle motion and illumination.

Recent work has revisited these ideas at the level of plane- and box-type models. E.g., Wang et al. (2010) estimate the geometric layout of walls in an indoor setting, segmenting out the clutter. Similarly, Hedau et al. (2010) estimate the layout of a room and reason about the locations of the bed as a box in the room. For indoor settings it has even been attempted to recover physical support relations, based on RGB-D data (Silberman et al., 2012). For fairly generic outdoor scenes, physical support, volumetric constraints and occlusions have also been included, still using boxes as object models (Gupta et al., 2010). It has also been observed that object detections carry information about 3D surface orientations, such that they can be jointly estimated even from a single image (Hoiem et al., 2008).

All the works indicate that even coarse 3D reasoning allows one to better guess the

(pseudo-)3D layout of a scene, while at the same time improving 2D recognition. Together with the above-mentioned strength of fine-grained shape models when it comes to occlusion and viewpoint, this is in our view a compelling reason to add 3D contextual constraints also to those fine-grained models.

Quantitative scene representations. A different type of methods also includes scene-level reasoning, but is tailored to specific applications and is more quantitative in nature. Most works in this direction target autonomous navigation, hence precise localization of reachable spaces and obstacles is important. Recent works for the autonomous driving scenario include: (Ess et al., 2009), in which multi-pedestrian tracking is done in 3D based on stereo video, and (Geiger et al., 2011; Wojek et al., 2013), both aiming for advanced scene understanding including multi-class object detection, 3D interaction modeling, as well as semantic labeling of the image content, from monocular input. Viewpoint estimates from semantic recognition can also be combined with interest point detection to improve camera pose and scene geometry even across wide baselines (Bao and Savarese, 2011). For indoor settings, a few recent papers also employ detailed object representations to support scene understanding (Del Pero et al., 2013), try to exploit frequently co-occurring object poses (Choi et al., 2013), and even supplement geometry and appearance constraints with affordances to better infer scene layout (Zhao and Zhu, 2013).

5.4 3D Object Model

We commence by introducing the fine-grained 3D object model that lies at the core of our approach. Its extension to entire multi-object scenes will be discussed in Section 5.5. By modeling an object class at the fine level of detail of individual wireframe vertices the object model provides the basis for reasoning about object extent and occlusion relations with high fidelity. To that end, we lift the pseudo-3D object model that we developed in Zia et al. (2013) to metric 3D space, and combine it with the explicit representation of likely occlusion patterns from Zia et al. (2013). Our object representation then comprises a model of global object geometry (Section 5.4.1), local part appearance (Section 5.4.2), and an explicit representation of occlusion patterns (Section 5.4.3). Additionally, the object representation also includes a grouping of local parts into semi-local part configurations (Section 5.4.4), which will be used to initialize the model during inference (Section 5.5.3). We depict the 3D object representation in Figure 5.2.

5.4.1 Global Object Geometry

We represent an object class as a deformable 3D wireframe, as in the classical “active shape model" formulation (Cootes et al., 1995). The vertices of the wireframe are defined manually, and wireframe exemplars are collected by annotating a set of 3D CAD models (*i.e.* , selecting corresponding vertices from their triangle meshes). Principal Component Analysis (PCA) is applied to obtain the mean configuration of vertices in 3D as well as the principal modes of their relative displacement. The final geometric object model then consists of the mean wireframe $\boldsymbol{\mu}$ plus the m principal component directions $\mathbf{p}_j$ and corresponding standard deviations σ_j, where $1 \leq j \leq m$. Any 3D wireframe $\mathbf{X}$ can thus be represented, up to some residual $\boldsymbol{\epsilon}$, as a linear combination of r principal components

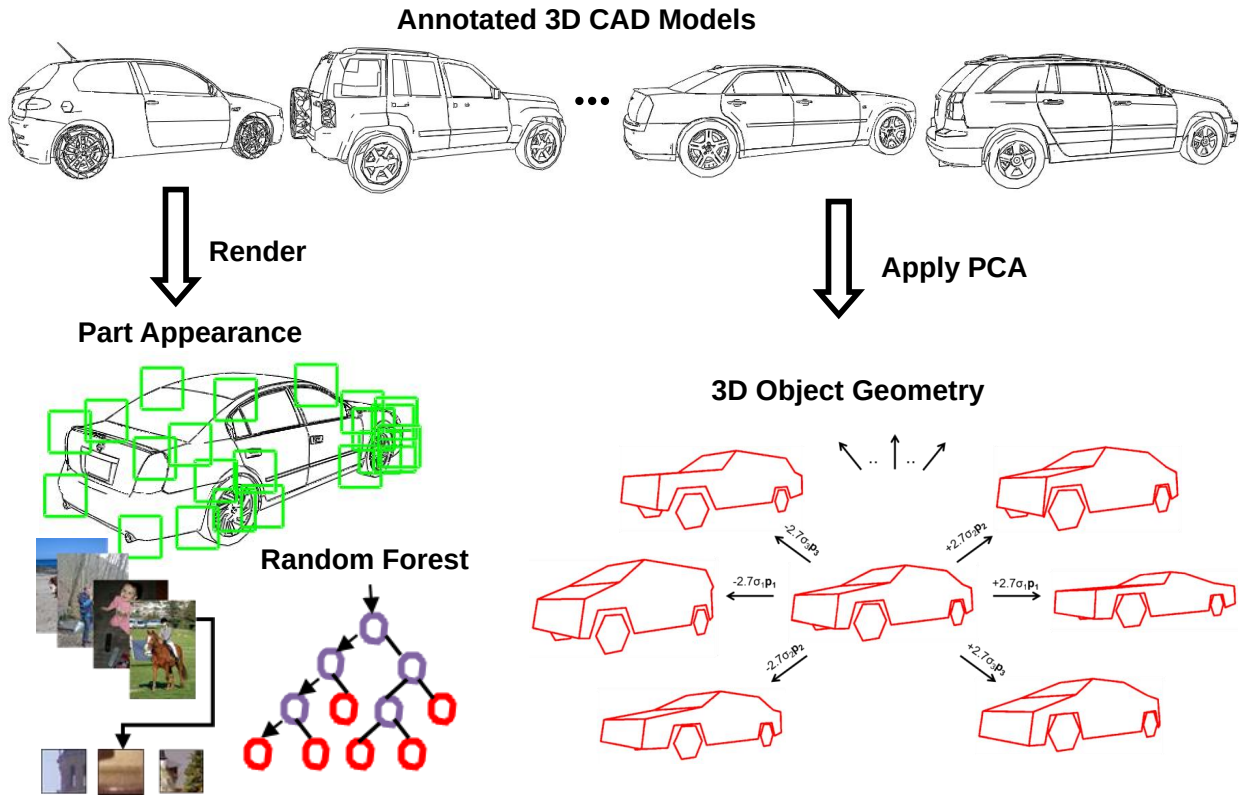


Figure 5.2: 3D Object Model.

with geometry parameters $\mathbf{s}$, where s_k is the weight of the k^{th} principal component:

$$\mathbf{X}(\mathbf{s}) = \boldsymbol{\mu} + \sum_{k=1}^{r} s_k \sigma_k \mathbf{p}_k + \boldsymbol{\epsilon} \tag{5.1}$$

Unlike the earlier Zia et al. (2013), the 3D CAD models are scaled according to their real world metric dimensions. [1]The resulting metric PCA model hence encodes physically meaningful scale information in world units, that allow one to assign absolute 3D positions to object hypotheses (given known camera intrinsics).

5.4.2 Local Part Appearance

We establish the connection between the 3D geometric object model (Section 5.4.1) and an image by means of a set of *parts*, one for each wireframe vertex. For each part, a multi-view appearance model is learned, by generating from training patches with non-photorealistic rendering of 3D CAD models from a large number of different viewpoints (Stark et al., 2010), and training a sliding-window detector on these patches.

Specifically, we encode patches around the projected locations of the annotated parts ($\approx$ 10% in size of the full object width) as dense shape context features (Belongie et al.,

[1]While in the earlier work they were scaled to the same size, so as to keep the deformations from the mean shape small.

2000). We learn a multi-class Random Forest classifier where each class represents the multi-view appearance of a particular part. We also dedicate a class trained on background patches, combining random real image patches with rendered non-part patches to avoid classifier bias. Using synthetic renderings for training allows us to densely sample the relevant portion of the viewing sphere with minimal annotation effort (one time labeling of part locations on 3D CAD models, *i.e.* no added effort in creating the shape model).

5.4.3 Explicit Occluder Representation

The 3D wireframe model allows one to represent partial occlusion at the level of individual parts: each part has an associated binary variable that stores whether the part is visible or occluded. Note that, in theory, this results in a exponential number of possible combinations of occluded and unoccluded parts, hindering efficient inference over occlusion states. We therefore take advantage of the fact that partial occlusion is not entirely random, but tends to follow re-occurring patterns that render certain joint occlusion states of multiple parts more likely than others (Pepik et al., 2013): the joint occlusion state depends on the shape of the occluding physical object(s).

Here we approximate the shapes of (hypothetical) occluders as a finite set of occlusion masks, following (Kwak et al., 2011; Zia et al., 2013). This set of masks constitutes a (hard) non-parameteric prior over possible occlusion patterns. The set is denoted by $\{a_i\}$, and for convenience we denote the empty mask which leaves the object fully visible by a_0. We sample the set of occlusion masks regularly from a generative model, by sliding multiple boxes across the mask in small spatial increments (the parameters of those boxes are determined empirically). Figure 5.3(b) shows a few out of the total 288 masks in our set, with the blue region representing the occluded portion of the object (car). The collection is able to capture different modes of occlusion, for example truncation by the image border (Figure 5.8(d), first row), occlusion in the middle by a post or tree (Figure 5.8(d), 2nd row), or occlusion of only the lower parts from one side (Figure 5.8(d), third row).

Note that the occlusion mask representation is independent of the cause of occlusion, and allows to uniformly treat occlusions that arise from *(i)* self occlusion (a part is occluded by a wireframe face of the same object), *(ii)* occlusion by another object that is part of the same scene hypothesis (a part is occluded by a wireframe face of another object), *(iii)* occlusion by an unknown source (a part is occluded by an object that is not part of the same scene hypothesis, or image evidence is missing).

5.4.4 Semi-Local Part Configurations

In the context of people detection and pose estimation, it has been realized that individual body parts are hard to accurately localize, because they are small and often not discriminative enough in isolation (Bourdev and Malik, 2009). Instead, it has proved beneficial to train detectors that span multiple parts appearing in certain poses (termed “poselets”), seen from a certain viewpoint, and selecting the ones that exhibit high discriminative power against background on a validation set. In line with these findings, we introduce

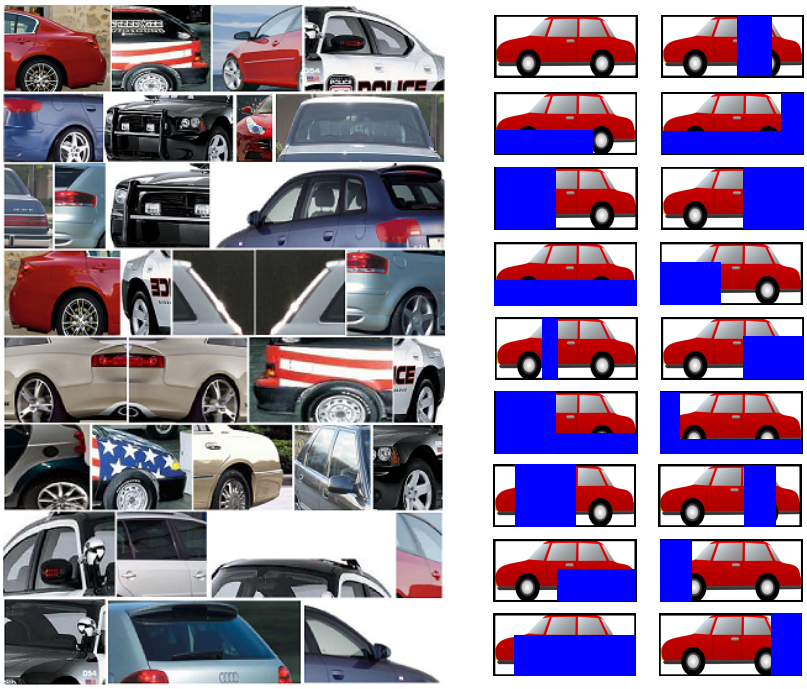


Figure 5.3: (a) Individual training examples for a few part $configurations$, (b) example occlusion masks.

the notion of part *configurations*, *i.e.* semi-local arrangements of a number of parts, seen from a specific viewpoint, that are adjacent (in terms of wireframe topology). Some examples are depicted in Figure 5.3(a)). These configurations provide more reliable evidence for each of the constituent parts than individual detectors. We use detectors for different configurations to find promising 2D bounding boxes and viewpoint estimates, as initializations for fitting the fine-grained 3D object models.

Specifically, we define certain configurations of adjacent parts, with different degrees of occlusion. Some configurations cover the full car, whereas others only span a part of it (down to $\approx 20\%$ of the full object). We then train a bank of single component DPM detectors, one for each configuration, in order to ensure high recall and a large number of object hypotheses to choose from. At test time, activations of these detectors are merged together through agglomerative clustering to form full object hypothesis, in the spirit of the poselet framework (Bourdev and Malik, 2009). For training, we utilize a set of images labeled at the level of individual parts, and with viewpoint labels from a small discrete set (in our experiments 8 equally spaced viewpoints). All the objects in these images are fully visible. Thus, we can store the relative scale and bounding box center offsets, w.r.t. the full object bounding box, for the part-configuration examples. When detecting potentially occluded objects in a test image, the activations of all configuration detectors predict a full object bounding box and a (discrete) pose.

Next we recursively merge nearby (in $x, y, scale$) activations that have the same viewpoint. Merging is accomplished by averaging the predicted full object bounding box corners, and assigning it the highest of the detection scores. After this agglomerative clustering has terminated all clusters above a fixed detection score are picked as legitimate objects.

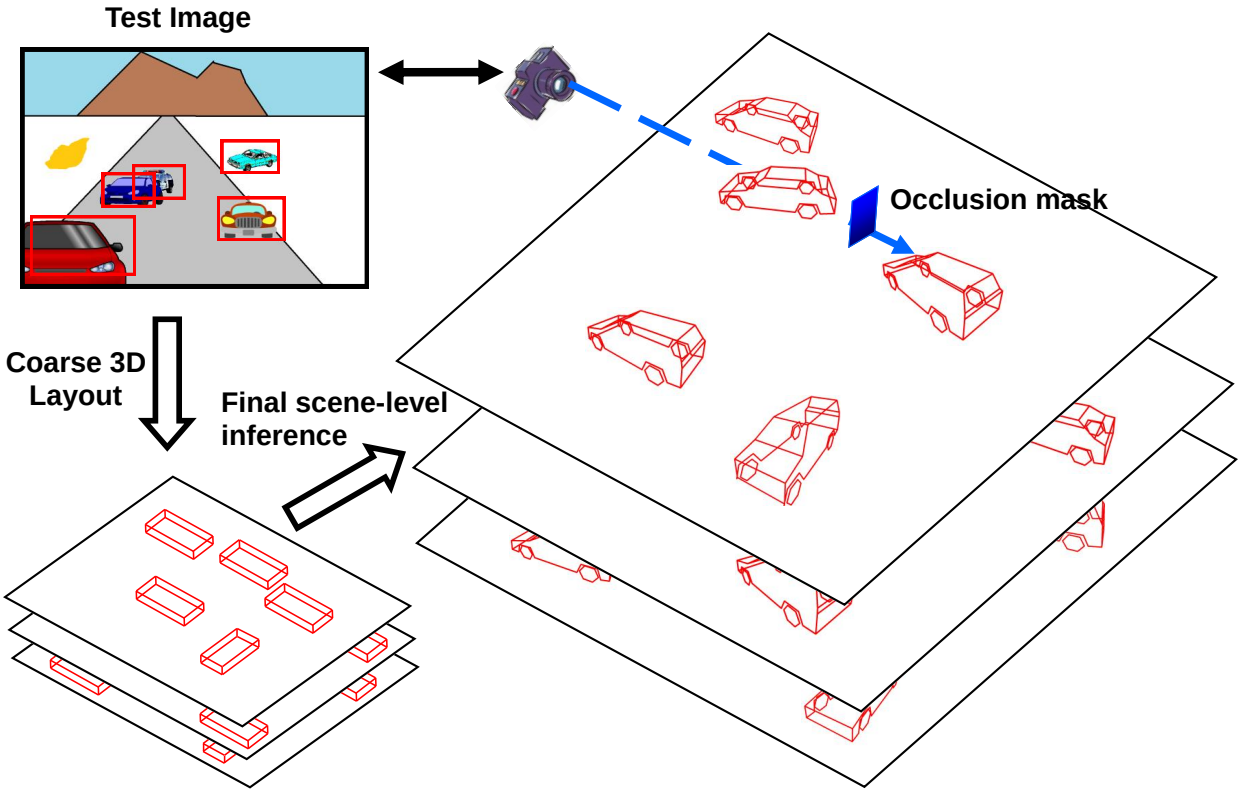


Figure 5.4: 3D Scene Model.

Thus we obtain full object bounding box predictions (even for partially visible objects), along with an approximate viewpoint.

5.5 3D Scene Model

We proceed by extending the single object model of Section 5.4 to entire scenes, where we can jointly reason about multiple objects and their geometric relations, placing them on a common ground plane and taking into account mutual occlusions. As we will show in the experiments (Section 5.6), this joint modeling can lead to significant improvements in terms of 3D object localization and pose estimation compared to separately modeling individual objects. It is enabled by a joint scene hypothesis space (Section 5.5.1), governed by a probabilistic formulation that scores hypotheses according to their likelihood (Section 5.5.2), and an efficient approximate inference procedure for finding plausible scenes (Section 5.5.3). The scene model is schematically depicted in Figure 5.4.

5.5.1 Hypothesis Space

Our 3D scene model comprises a common ground plane and a set of 3D deformable wireframes with corresponding occlusion masks (Section 5.4). Note that this hypothesis space is more expressive than the 2.5 D representations used by previous work (Ess et al., 2009; Meger et al., 2011; Wojek et al., 2013), as it allows reasoning about locations,

shapes, and interactions of objects, at the level of individual 3D wireframe vertices and faces.

Common ground plane. In the full system, we constrain all the object instances to lie on a common ground plane, as often done for street scenes. This assumption usually holds and drastically reduces the search space for possible object locations (2 degrees of freedom for translation and 1 for rotation, instead of $3+3$). Moreover, the consensus for a common ground plane stabilizes 3D object localization. We parametrize the ground plane with the pitch and roll angles relative to the camera frame, $\boldsymbol{\theta}_{gp} = (\theta_{pitch}, \theta_{roll})$. The height q_y of the camera above ground is assumed known and fixed.

Object instances. Each object in the scene is an instance of the 3D wireframe model described in Section 5.4.1. An individual instance $\mathbf{h}^\beta = (\mathbf{q}, \mathbf{s}, a)$ comprises 2D translation and azimuth $\mathbf{q} = (q_x, q_z, q_{az})$ relative to the ground plane, shape parameters $\mathbf{s}$, and an occlusion mask a.

Explicit occlusion model. As detailed in Section 5.4.3, we represent occlusions on an object instance by selecting an occluder mask out of a pre-defined set $\{a_i\}$, which in turn determines the binary occlusion state of all parts. That is, the occlusion state of part j is given by an indicator function $o_j(\boldsymbol{\theta}_{gp}, q_{az}, \mathbf{s}, a)$, with $\boldsymbol{\theta}_{gp}$ the ground plane parameters, q_{az} the object azimuth, $\mathbf{s}$ the object shape, and a the occlusion mask. Since all object hypotheses reside in the same 3D coordinate system, mutual occlusions can be derived deterministically from their depth ordering (Figure 5.4): we cast rays from the camera center to each wireframe vertex of all other objects, and record intersections with faces of any other object as an appropriate occlusion mask. Accordingly, we write $\Gamma\big(\{\mathbf{h^1}, \mathbf{h^2}, \ldots, \mathbf{h^n}\} \backslash \mathbf{h}^\beta, \mathbf{h}^\beta, \boldsymbol{\theta}_{gp}\big)$, *i.e.* the operator Γ returns the index of the occlusion mask for $\mathbf{h}^\beta$ as a function of the other objects in a given scene estimate.

5.5.2 Probabilistic Formulation

All evidence in our model comes from object part detection, and the prior for allowable occlusions is given by per-object occlusion masks and relative object positions (Section 5.5.1).

Object likelihood. The likelihood of an object being present at a particular location in the scene is measured by responses of a bank of (viewpoint-independent) sliding-window part detectors (Section 5.4.2), evaluated at projected image coordinates of the corresponding 3D wireframe vertices.[2] The likelihood $\mathcal{L}(\mathbf{h}^\beta, \boldsymbol{\theta}_{gp})$ for an object $\mathbf{h}^\beta$ standing on the ground plane $\boldsymbol{\theta}_{gp}$ is the sum over the responses of all visible parts, with a constant likelihood for occluded parts:

$$\mathcal{L}(\mathbf{h}^\beta, \boldsymbol{\theta}_{gp}) = \max_{\boldsymbol{\varsigma}} \left[\frac{\sum_{j=1}^{m} \big(\mathcal{L}_v + \mathcal{L}_o\big)}{\sum_{j=1}^{m} o_j(\boldsymbol{\theta}_{gp}, q_{az}, \mathbf{s}, a_0)} \right]. \tag{5.2}$$

The denominator normalizes for the varying number of self-occluded parts at different viewpoints. $\mathcal{L}_v$ is the evidence (pseudo log-likelihood) $S_j(\boldsymbol{\varsigma}, \mathbf{x}_j)$ for part j if it is visible,

[2] In practice this amounts to a look-up in the precomputed response maps.

found by looking up the detection score at image location $\mathbf{x}_j$ and scale $\boldsymbol{\varsigma}$, normalized with the background score $S_b(\boldsymbol{\varsigma}, \mathbf{x}_j)$ as in (Villamizar et al., 2011). $\mathcal{L}_o$ assigns a fixed likelihood c to an occluded part:

$$\mathcal{L}_v = o_j(\boldsymbol{\theta}_{gp}, q_{az}, \mathbf{s}, a) \log \frac{S_j(\boldsymbol{\varsigma}, \mathbf{x}_j)}{S_b(\boldsymbol{\varsigma}, \mathbf{x}_j)} \ , \tag{5.3}$$

$$\mathcal{L}_o = \big(o_j(\boldsymbol{\theta}_{gp}, q_{az}, \mathbf{s}, a_0) - o_j(\boldsymbol{\theta}_{gp}, q_{az}, \mathbf{s}, a)\big) c \ . \tag{5.4}$$

Scene-level likelihood. To score an entire scene we combine object hypotheses and ground plane into a scene hypothesis $\psi = \{q_y, \boldsymbol{\theta}_{gp}, \mathbf{h^1}, ..., \mathbf{h^n}\}$. The likelihood of a complete scene is then the sum over all object likelihoods, such that the objective for scene interpretation becomes:

$$\hat{\psi} = \arg\max_{\psi} \left[\sum_{\beta=1}^{n} \mathcal{L}(\mathbf{h}^\beta, \boldsymbol{\theta}_{gp}) \right] . \tag{5.5}$$

Note, the domain $Dom\big(\mathcal{L}\big(\mathbf{h}^\beta, \boldsymbol{\theta}_{gp}\big)\big)$ must be limited such that the occluder mask a^β of an object hypothesis $\mathbf{h}^\beta$ is dependent on relative poses of all the objects in the scene: an object hypothesis $\mathbf{h}^\beta$ can only be assigned occlusion masks a_i which respect object-object occlusions—*i.e.* at least all the vertices covered by $\Gamma\big(\{\mathbf{h^1}, \mathbf{h^2}, \ldots, \mathbf{h^n}\} \backslash \mathbf{h}^\beta, \mathbf{h}^\beta, \boldsymbol{\theta}_{gp}\big)\big)$ must be covered, even if a different mask would give a higher objective value. Also note that the ground plane in our current implementation is a hard constraint—objects off the ground are impossible in our parameterization (except for experiments in which we "turn off" the ground plane for comparison).

5.5.3 Inference

The objective function in Eqn. 5.5 is high-dimensional, highly non-convex, and not smooth (due to the binary occlusion states). Note that deterministic occlusion reasoning potentially introduces dependencies between all pairs of objects, and the common ground plane effectively ties all other variables to the ground plane parameters $\boldsymbol{\theta}_{gp}$. In order to still do approximate inference and reach strong local maxima of the likelihood function, we have designed an inference scheme that proceeds in stages, lifting an initial 2D guess (*Initialization*) about object locations to a coarse 3D model (*Coarse 3D Geometry*), and refining that coarse model into a final collection of consistent 3D shapes (*Final scene-level inference, Occlusion Reasoning*).

Initialization. We initialize the inference from coarse 2D bounding box pre-detections and corresponding discrete viewpoint estimates (Section 5.4.4), keeping all pre-detections above a confidence threshold. Note that this implicitly determines the maximum number of objects that will be considered in the scene hypothesis under consideration.

Coarse 3D geometry. Since we reason in a fixed, camera-centered 3D coordinate frame, the initial detections can be directly lifted to 3D space, by casting rays through 2D bounding box centers and instantiating objects on these rays, such that their reprojections are consistent with the 2D boxes and discrete viewpoint estimates, and reside on a common ground plane. In order to avoid discretization artifacts, we then refine the lifted

object boxes by imputing the mean object shape and performing a grid search over ground plane parameters and object translation and rotation (azimuth). In this step, rather than commiting to a single scene-level hypothesis, we retain many candidate hypotheses (*scene particles*) that are consistent with the 2D bounding boxes and viewpoints of the pre-detections within some tolerance.

Occlusion reasoning. We combine two different methods to select an appropriate occlusion mask for a given object, *(i)* deterministic occlusion reasoning, and *(ii)* occlusion reasoning based on (the absence of) part evidence.

(i) Since by construction we recover the 3D locations and shapes of multiple objects in a common frame, we can calculate whether a certain object instance is occluded by any other modeled object instance in our scene. This is calculated efficiently by casting rays to all (not self-occluded) vertices of the object instance, and checking if a ray intersects any other object in its path before reaching the vertex. This deterministically tells us which parts of the object instance are occluded by another modeled object in the scene, allowing us to choose an occluder mask that best represents the occlusion (overlaps the occluded parts). To select the best mask we search through the entire set of occluders to maximize the number of parts with the correct occlusion label, with greater weight on the occluded parts (in the experiments, twice as much as for visible parts).

(ii) For parts not under deterministic occlusion, we look for missing image evidence (low part detection scores for multiple adjacent parts), guided by the set of occluder masks. Specifically, for a particular wireframe hypothesis, we search through the set of occluder masks to maximize the summed part detection scores (obtained from the Random Forest classifier, Section 5.4.2), replacing the scores for parts behind the occluder by a constant (low) score c. Especially in this step, leveraging local context in the form of occlusion masks stabilizes individual part-level occlusion estimates, which by themselves are rather unreliable because of the noisy evidence.

Final scene-level inference. Finally, we search a good local optimum of the scene objective function (Eqn. 5.5) using an iterative stochastic optimization scheme shown in Algorithm 3. The procedure is based on block coordinate descent to decouple shape and viewpoint variables, combined with ideas from smoothing-based optimization (Leordeanu and Hebert, 2008). As the space of ground planes is already well-covered by the set of multiple scene particles (in our experiments 250), we keep the ground plane parameters of each particle constant. This stabilizes the optimization.

Each particle is iteratively refined in two steps: first, the shape and viewpoint parameters of all objects are updated, by testing many random perturbations around the current values and keeping the best one. The random perturbations follow a normal distribution that is adapted in a data-driven fashion (Leordeanu and Hebert, 2008). Then, object occlusions are recomputed and occlusions by unmodeled objects are updated, by exhaustive search over the set of possible masks. For each scene particle these two update steps are iterated, and the particle with the highest objective value ψ forms our MAP estimate.

Given: Scene particle ψ': initial objects $\mathbf{h}^\beta = (\mathbf{q}^\beta, \mathbf{s}^\beta, a^\beta)$, $\beta = 1 \dots n$; fixed $\boldsymbol{\theta}_{gp}$; $a^\beta = a_0$ (all objects fully visible)
for *fixed number of iterations* **do**
  1. for $\beta = 1 \dots n$ **do**
    draw samples $\{\mathbf{q}_j^\beta, \mathbf{s}_j^\beta\}^{j=1..m}$ from a Gaussian $\mathcal{N}(\mathbf{q}^\beta, \mathbf{s}^\beta; \Sigma^\beta)$ centered at current values;
    update $\mathbf{h}^\beta = \text{argmax}_j \; \mathcal{L}\big(\mathbf{h}^\beta(\mathbf{q}_j^\beta, \mathbf{s}_j^\beta, a^\beta), \boldsymbol{\theta}_{gp}\big)$
  end
  2. for $\beta = 1 \dots n$ **do**
    update occlusion mask (exhaustive search)
    $a^\beta = \text{argmax}_j \;\; \mathcal{L}\big(\mathbf{h}^\beta(\mathbf{q}^\beta, \mathbf{s}^\beta, a_j), \boldsymbol{\theta}_{gp}\big)$
  end
  3. Recompute sampling variance Σ^β of Gaussians (Leordeanu and Hebert, 2008)
end

Algorithm 3: Inference run for each scene particle.

5.6 Experiments

In this section, we extensively analyze the performance of our fine-grained 3D scene model, focusing on its ability to derive 3D estimates from a single input image (with known camera intrinsics). To that end, we evaluate object localization in 3D metric space (Section 5.6.4) as well as 3D pose estimation (Section 5.6.4) on the challenging KITTI dataset (Geiger et al., 2012) of street scenes. In addition, we analyze the performance of our model w.r.t. part-level occlusion prediction and part localization in the 2D image plane (Section 5.6.5). In all experiments, we compare the performance of our full model with stripped-down variants as well as appropriate baselines, to highlight the contributions of different system components to overall performance.

5.6.1 Dataset

In order to evaluate our approach for 3D layout estimation from a single view, we require a dataset with 3D annotations. We thus turn to the KITTI *3D object detection and orientation estimation* benchmark dataset (Geiger et al., 2012) as a testbed for our approach, since it provides challenging images of realistic street scenes with varying levels of occlusion and clutter, but nevertheless controlled enough conditions for thorough evaluations. It consists of around $7,500$ training and $7,500$ test images of street scenes captured from a moving vehicle and comes with labeled 2D and 3D object bounding boxes and viewpoints (generated with the help of a laser scanner).

Test set. Since annotations are only made publicly available on the training set of KITTI, we utilize a portion of this training set for our evaluation. We choose only images with multiple cars that are large enough to identify parts, and manually annotate all cars in this subset with 2D part locations and part-level occlusion labels. Specifically, we pick every 5th image from the training set with at least two cars with height greater than 75 pixels. This gives us 260 test images with 982 cars in total, of which 672 are partially

occluded, and 476 are severely occluded. Our selection shall ensure that while being biased towards more complex scenes, we still sample a representative portion of the dataset.

Training set. We use two different kinds of data for training our model, *(i)* synthetic data in the form of rendered CAD models, and *(ii)* real-world training data. *(i)* We utilize 38 commercially available 3D CAD models of cars for learning the object wireframe model as well as for learning viewpoint-invariant part appearances, (c.f. Zia et al., 2013). Specifically, we render the 3D CAD models from 72 different azimuth angles (5° steps) and 2 elevation angles (7.5° and 15° above the ground), densely covering the relevant part of the viewing sphere, using the non-photorealistic style of Stark et al. (2010). Rendered part patches serve as positive part examples, randomly sampled image patches as well as non-part samples from the renderings serve as negative background examples to train the multi-class Random Forest classifier. The classifier distinguishes 37 classes (36 parts and 1 background class), using 30 trees with a maximum depth of 13. The total number of training patches is $162,000$, split into $92,000$ part and $70,000$ background patches. *(ii)* We train 118 part configuration detectors (single component DPMs) labeled with discrete viewpoint, 2D part locations and part-level occlusion labels on a set of $1,000$ car images downloaded from the internet and 150 images from the KITTI dataset (none of which are part of the test set). In order to model the occlusions, we semi-automatically define a set of 288 occluder masks, the same as in Zia et al. (2013).

5.6.2 Object Pre-Detection

As a sanity check, we first verify that our 2D pre-detection (Section 5.4.4) matches the state-of-the-art. To that end we evaluate a standard 2D bounding box detection task according to the PASCAL VOC criterion ($> 50\%$ intersection-over-union between predicted and ground truth bounding boxes). As normally done we restrict the evaluation to objects of a certain minimum size and visibility. Specifically, we only consider cars > 50 pixels in height which are at least 20% visible. The minimum size is slightly stricter than the 40 pixels that Geiger et al. (2012) use for the dataset (since we need to ensure enough support for the part detectors), whereas the occlusion threshold is much more lenient than their 80% (since we are specifically interested in occluded objects).

Results. We compare our bank of single component DPM detectors to the original deformable part model (Felzenszwalb et al., 2010), both trained on the same training set (Section 5.6.1). Precision-recall curves are shown in Figure 5.6. We observe that our detector bank (green curve, 57.8% AP) in fact performs slightly better than the original DPM (red curve, 57.3% AP). In addition, it delivers coarse viewpoint estimates and rough part locations that we can leverage for initializing our scene-level inference (Section 5.5.3).

5.6.3 Model Variants and Baselines

We compare the performance of our full system with a number of stripped down variants in order to quantify the benefit that we get from each individual component. We consider the following variants:

	full dataset		occ >0 parts		occ >3 parts	
	<1m	<1.5m	<1m	<1.5m	<1m	<1.5m
Figure 5.5 plot	*(a)*	*(b)*			*(c)*	*(d)*
(i) fg	23%	35%	22%	31%	23%	32%
(ii) fg+so	26%	37%	23%	33%	27%	36%
(iii) fg+do	25%	37%	26%	35%	27%	38%
(iv) fg+gp	40%	53%	40%	52%	38%	49%
(v) fg+gp+do+so	**44%**	**56%**	**44%**	**55%**	**43%**	**60%**
(vi) Zia et al. (2013)	—	—	—	—	—	—
(vii) coarse	21%	37%	21%	40%	20%	42%
(viii) coarse+gp	35%	54%	28%	48%	27%	47%

Table 5.1: 3D localization accuracy: percentage of cars correctly localized within 1 and 1.5 meters of ground truth.

(i) fg: the basic version of our fine-grained 3D object model, without ground plane, searched occluder or deterministic occlusion reasoning; this amounts to independent modeling of the objects in a common, metric 3D scene coordinate system. *(ii)* fg+so: same as (i) but with searched occluder to represent occlusions caused by unmodeled scene elements. *(iii)* fg+do: same as (i) but with deterministic occlusion reasoning between multiple objects. *(iv)* fg+gp: same as (i), but with common ground plane. *(v)* fg+gp+do+so: same as (i), but with all three components, common ground plane, searched occluder, and deterministic occlusion turned on. *(vi)* the earlier pseudo-3D shape model (Zia et al., 2013), with probabilistic occlusion reasoning; this uses essentially the same object model as (ii), but learns it from examples scaled to the *same* size rather than the *true* size, and fits the model in 2D $(x, y, scale)$-space rather explicitly recovering a 3D scene interpretation.
We also compare our representation to two different baselines, *(vii)* coarse: a scene model consisting of 3D bounding boxes rather than detailed cars, corresponding to the coarse 3D geometry stage of our pipeline (Section 5.5.3); and *(viii)* coarse+gp: like *(vii)* but with a common ground plane for the bounding boxes. Specifically, during the coarse grid search we choose the 3D bounding box hypothesis whose 2D projection is closest to the corresponding pre-detection 2D bounding box.

5.6.4 3D Evaluation

Having verified that our pre-detection stage is competitive and provides reasonable object candidates in the image plane, we now move on to the more challenging task of estimating the 3D location and pose of objects from monocular images (with known camera intrinsics). As we will show, the fine-grained representation leads to significant performance improvements over a standard baseline that considers only 3D bounding boxes, on both tasks.

3D Object Localization

Protocol. We measure 3D localization performance by the fraction of detected object centroids that are correctly localized up to deviations of 1, and 1.5 meters. These

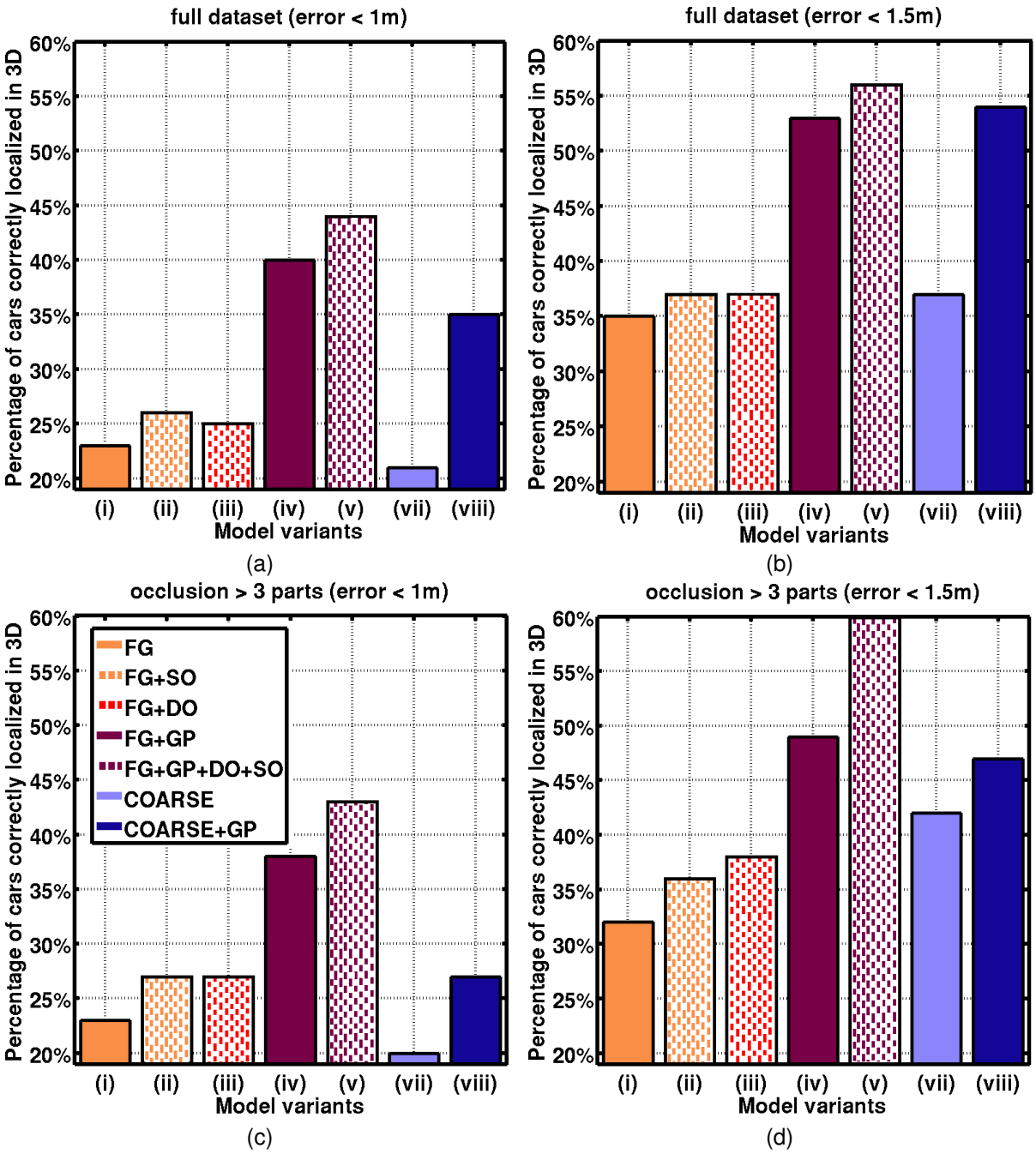


Figure 5.5: 3D localization accuracy: percentage of cars correctly localized within 1 (a,c) and 1.5 (b,d) meters of ground truth, on all (a,b) and occluded (c,d) cars.

thresholds may seem rather strict for the viewing geometry of KITTI, but in our view larger tolerances make little sense for cars with dimensions $\approx 4.0 \times 1.6$ meters.

In line with existing studies on pose estimation, we base the analysis on true positive (TP) initializations that meet the PASCAL VOC criterion for 2D bounding box overlap and whose coarse viewpoint estimates lie within 45° of the ground truth, thus excluding failures of pre-detection. We perform the analysis for three settings (Table 5.1): *(i)* over

our full testset (517 of 982 TPs); *(ii)* only over those cars that are partially occluded, *i.e.* 1 or more of the parts that are not self-occluded by the object are not visible (234 of 672 TPs); and *(iii)* only those cars that are severely occluded, *i.e.* 4 or more parts are not visible (113 of 476 TPs). Figure 5.5 visualizes selected columns of Table 5.1 as bar plots to facilitate the comparison.

Results. In Table 5.1 and Fig 5.5, we first observe that our full system (fg+gp+do+so, dotted dark red) is the top performer for all three occlusion settings and both localization error thresholds, localizing objects with 1 m accuracy in $43-44\%$ of the cases and with 1.5 m accuracy in 55–60% of the cases. Figure 5.8 visualizes some examples of our full system fg+gp+do+so vs. the stronger baseline coarse+gp.

Second, the basic fine-grained model fg (orange) outperforms coarse (light blue) by 1–3 percent points (pp) corresponding to a relative improvement of 4–13% at 1 m accuracy. The gains increase by a large margin when adding a ground plane: fg+gp (dark red) outperforms coarse+gp (dark blue) by 5–12 pp (13–43%) at 1 m accuracy. In other words, cars are not 3D boxes. Modeling their detailed shape and pose yields better scene descriptions, with and without ground plane constraint. The results at 1.5 m are less clear-cut. It appears that from badly localized initializations just inside the 1.5 m radius, the final inference sometimes drifts into incorrect local minima outside of 1.5 m.

Third, modeling fine-grained occlusions either independently (fg+so, dotted orange) or deterministically across multiple objects (fg+do, dotted red) brings marked improvements on top of fg alone. At 1 m they outperform fg by 1–4 pp (2–15%) and by 2–4 pp (7–19%), respectively. We get similar improvements at 1.5 m, with fg+so and fg+do outperforming fg by 2–4 pp (4–14%), and 2–6 pp (4–19%) respectively. Not surprisingly, the performance boost is greater for the occluded cases, and both occlusion reasoning approaches are in fact beneficial for 3D reasoning. Figure 5.9 visualizes some results with and without occlusion reasoning.

And last, adding the ground plane always boosts the performance for both the fg and coarse models, strongly supporting the case for joint 3D scene reasoning: at 1 m accuracy the gains are 15–18 pp (65–81%) for fg+gp vs. fg, and 7–14 pp (30–67%) for coarse+gp vs. coarse. Similarly, at 1.5 m accuracy we get 17–21 pp (51–68%) for fg+gp vs. fg, and 5–17 pp (10–47%) for coarse+gp vs. coarse. for qualitative results see Figure 5.10.

We obtain even richer 3D "reconstructions" by replacing wireframes with nearest neighbors from the database of 3D CAD models (Figure 5.11), accurately recognizing hatchbacks (a, e, f, i, j, l, u), sedans (b, o) and station wagons (d, p, v, w, x), as well as approximating the van (c, no example in database) by a station wagon.

Viewpoint Estimation

Beyond 3D location, 3D scene interpretation also requires the viewpoint of every object, or equivalently its orientation in metric 3D space. Many object classes are elongated, thus their orientation is valuable at different levels, ranging from low-level tasks such as detecting occlusions and collisions to high-level ones like enforcing long-range

	full dataset				occ >0 parts				occ >3 parts			
	<5°	<10°	3D err	2D err	<5°	<10°	3D err	2D err	<5°	<10°	3D err	2D err
(i) fg	44%	**69%**	**5°**	**4°**	41%	65%	6°	4°	35%	**58%**	**7°**	5°
(ii) fg+so	42%	66%	6°	**4°**	39%	62%	6°	4°	33%	53%	8°	5°
(iii) fg+do	**45%**	68%	**5°**	**4°**	44%	**66%**	6°	4°	36%	56%	**7°**	**4°**
(iv) fg+gp	41%	63%	6°	**4°**	40%	62%	6°	4°	36%	52%	8°	5°
(v) fg+gp+do+so	44%	65%	6°	**4°**	**47%**	65%	**5°**	**3°**	**44%**	55%	8°	**4°**
(vi) Zia et al. (2013)	-	-	-	6°	-	-	-	6°	-	-	-	6°
(vii) coarse	16%	38%	13°	9°	20%	41%	13°	6°	21%	40%	14°	9°
(viii) coarse+gp	25%	51%	10°	6°	27%	51%	10°	5°	23%	40%	14°	7°

Table 5.2: 3D viewpoint estimation accuracy (percentage of objects with less than 5° and 10° error) and median angular estimation errors (3D and 2D)

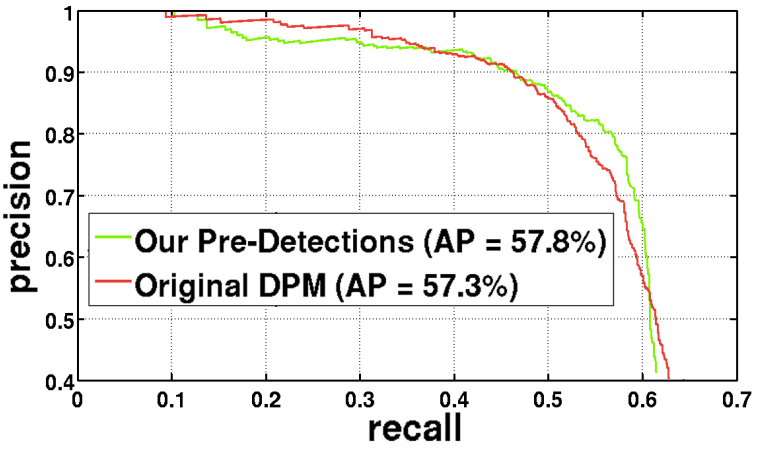


Figure 5.6: Object pre-detection performance.

regularities (*e.g.* cars parked at the roadside are usually parallel).

Protocol. We can evaluate object orientation (azimuth) in 2D image space as well as in 3D scene space. 2D viewpoint is the apparent azimuth of the object as seen in the image. The actual azimuth relative to a fixed scene direction (called 3D viewpoint), is calculated from the 2D viewpoint estimate and an estimate of 3D object location. We measure viewpoint estimation accuracy in two ways: as the percentage of detected objects for which the 3D angular error is below 5° or 10°, and as the median angular error between estimated and ground truth azimuth angle over detected objects, both in 3D and 2D.

Results. Table 5.2 shows the quantitative results, again comparing our full model and the different variants introduced in Section 5.6.3, and distinguishing between the full dataset and two subsets with different degrees of occlusion. In Figure 5.7 we plot the percentage of cars whose poses are estimated correctly up to different error thresholds, using the same color coding as Figure 5.5.

First, we observe that the full system fg+gp+do+so (dotted dark red) outperforms the best coarse model coarse+gp (dark blue) by significant margins of 19–21 pp and 14–15 pp

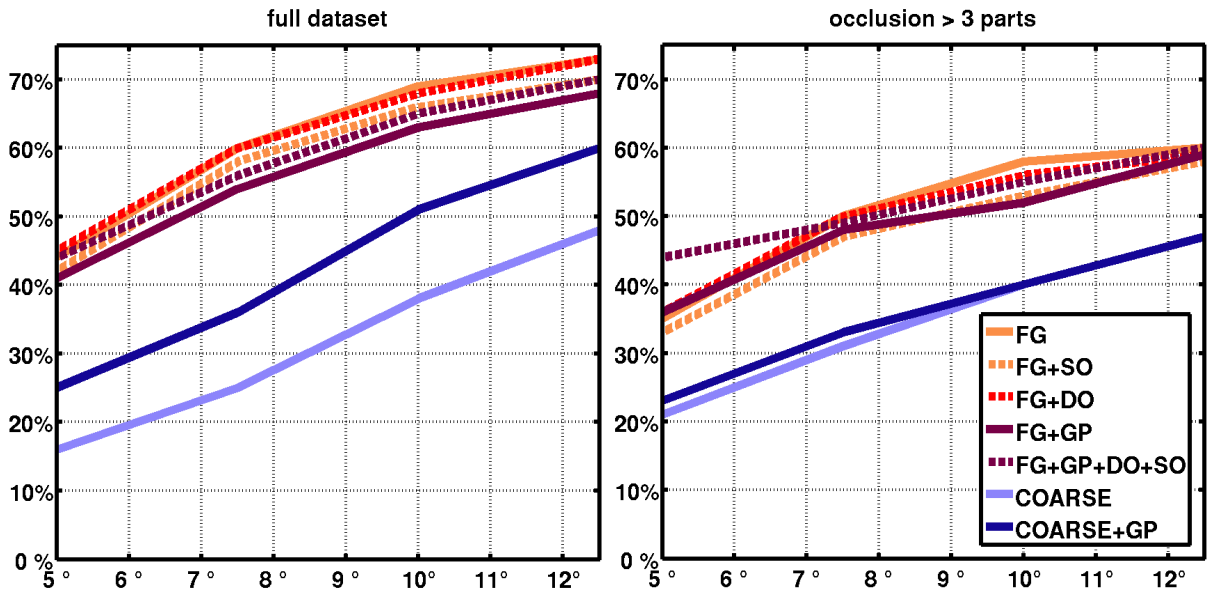


Figure 5.7: Percentage of cars with VP estimation error within x°.

at 5° and 10° errors respectively, improving the median angular error by 4°–6°.

Second, all fg models (shades of orange and red) deliver quite reliable viewpoint estimates with smaller differences in performance ($\leq$ 6 pp, or 1° median error) for 10° error, outperforming their respective coarse counterparts (shades of blue) by significant margins. Observe the clear grouping of curves in Figure 5.7. However, for the high accuracy regime ($\leq 5^\circ$ error), the full system fg+gp+do+so (dotted dark red) delivers the best performance for both occluded subsets, beating the next best combination fg+do (dotted red) by 3 pp and 8 pp, respectively.

Third, the ground plane helps considerably for the coarse models (shades of blue), improving by 9 pp for $\leq 5^\circ$ error, and 13 pp for $\leq 10^\circ$ over the full data set. Understandably, that gain gradually dissolves with increasing occlusion.

And fourth, we observe that in terms of median 2D viewpoint estimation error, our full system fg+gp+do+so outperforms the pseudo-3D model of (Zia et al., 2013) by 2°–3°, highlighting the benefit of reasoning in true metric 3D space.

5.6.5 2D Evaluation

While the objective of this work is to enable accurate localization and pose estimation in 3D (Section 5.6.4), we also present an analysis of 2D performance (part localization and occlusion prediction in the image plane), to put the work into context. Unfortunately, a robust measure to quantify how well the wireframe model fits the image data requires accurate ground truth 2D locations of even the occluded parts, which are not available. A measure used previously in Zia et al. (2013) is 2D part localization accuracy only evaluated for the visible parts, but we now find it to be biased, because fitting the model to just the visible parts leads to high accuracies on that measure, even if the overall fit is grossly

	full dataset		occ >0 parts		occ >3 parts	
	occlusion prediction accuracy	No. of cars with >70% parts	occlusion prediction accuracy	No. of cars with >70% parts	occlusion prediction accuracy	No. of cars with >70% parts
(i) fg	82%	69%	70%	**68%**	57%	43%
(ii) fg+so	87%	66%	80%	63%	77%	35%
(iii) fg+do	84%	70%	72%	67%	62%	**47%**
(iv) fg+gp	82%	68%	68%	67%	57%	46%
(v) fg+gp+do+so	**88%**	**71%**	82%	67%	79%	44%
(vi) Zia et al. (2013)	87%	64%	**84%**	61%	**84%**	32%
(vii) coarse	—	—	—	—	—	—
(viii) coarse+gp	—	—	—	—	—	—

Table 5.3: 2D accuracy. Part-level occlusion prediction accuracy and percentage of cars which have >70% parts accurately localized.

incorrect. We thus introduce a more robust measure below.

Protocol. We follow the evaluation protocol commonly applied for human body pose estimation and evaluate the number of correctly localized parts, using a relative threshold adjusted to the size of the reprojected car (20 pixels for a car of size 500×170 pixels, *i.e.* $\approx 4\%$ of the total length (c.f. Zia et al., 2013)). We use this threshold to determine the percentage of detected cars for which 70% or more of all (not self-occluded) parts are localized correctly, evaluated on cars for which at least 70% of the (not self-occluded) parts are visible according to ground truth. We find this measure to be more robust, since it favours sensible fits of the overall wireframe.

Further, we calculate the percentage of (not self-occluded) parts for which the correct occlusion label is estimated. For the model variants which do not use the occluder representation (fg and fg+gp), all candidate parts are predicted as visible.

Results. Table 5.3 shows the results for both 2D part localization and part-level occlusion estimation. We observe that our full system fg+gp+do+so is the highest performing variant over the full data set (88% part-level occlusion prediction accuracy and 71% cars with correct part localization). For the occluded subsets, the full system performs best among all fg models on occlusion prediction, whereas the results for part localization are less conclusive. An interesting observation is that methods that use 3D context (fg+gp+do+so, fg+gp, fg+do) consistently beat (fg+so), *i.e.* inferring occlusion is more brittle from (missing) image evidence alone than when supported by 3D scene reasoning.

Comparing the pseudo-3D baseline (Zia et al., 2013) and its proper metric 3D counterpart fg+so, we observe that, indeed, metric 3D improves part localization by 2–3 pp (despite inferior part-level occlusion prediction). In fact, all fg variants outperform Zia et al. (2013) in part localization by significant margins, notably fg+gp+do+so (6–12 pp).

On average, we note that there is only a weak (although still positive) correlation between 2D part localization accuracy and 3D localization performance (Section 5.6.4). In other words, whenever possible *3D reasoning should be evaluated in 3D space*, rather than in the 2D projection.[3]

5.7 Conclusion

We have approached the 3D scene understanding problem from the perspective of detailed deformable shape and occlusion modeling, jointly fitting the shapes of multiple objects linked by a common scene geometry (ground plane). Our results suggest that detailed representations of object shape are beneficial for 3D scene reasoning, and fit well with scene-level constraints between objects. By itself, fitting a detailed, deformable 3D model of cars and reasoning about occlusions resulted in improvements of 16–26% in object localization accuracy (number of cars localized to within 1m in 3D), over a baseline which just lifts objects' bounding boxes into the 3D scene. Enforcing a common ground plane for all 3D bounding boxes improved localization by 30–67%. When both aspects are combined into a joint model over multiple cars on a common ground plane, each with its own detailed 3D shape and pose, we get a striking 104–113% improvement in 3D localization compared to just lifting 2D detections, as well as a reduction of the median orientation error from 13° to 5°. We also find that the increased accuracy in 3D scene coordinates is not reflected in improved 2D localization of the shape model's parts, supporting our claim that 3D scene understanding should be carried out (and evaluated) in an explicit 3D representation.

An obvious limitation of the present system, to be addressed in future work, is that it only includes a single object category, and applies to the simple (albeit important) case of scenes with a dominant ground plane. In terms of technical approach it woud be desirable to develop a better and more efficient inference algorithm for the joint scene model. Finally, the bottleneck where most of the recall is lost is the 2D pre-detection stage. Hence, either better 2D object detectors are needed, or 3D scene estimation must be extended to run directly on entire images without initialization, which will require greatly increased robustness and efficiency.

[3] Note, there is no 3D counterpart to this part-level evaluation, since we see no way to obtain sufficiently accurate 3D part annotations.

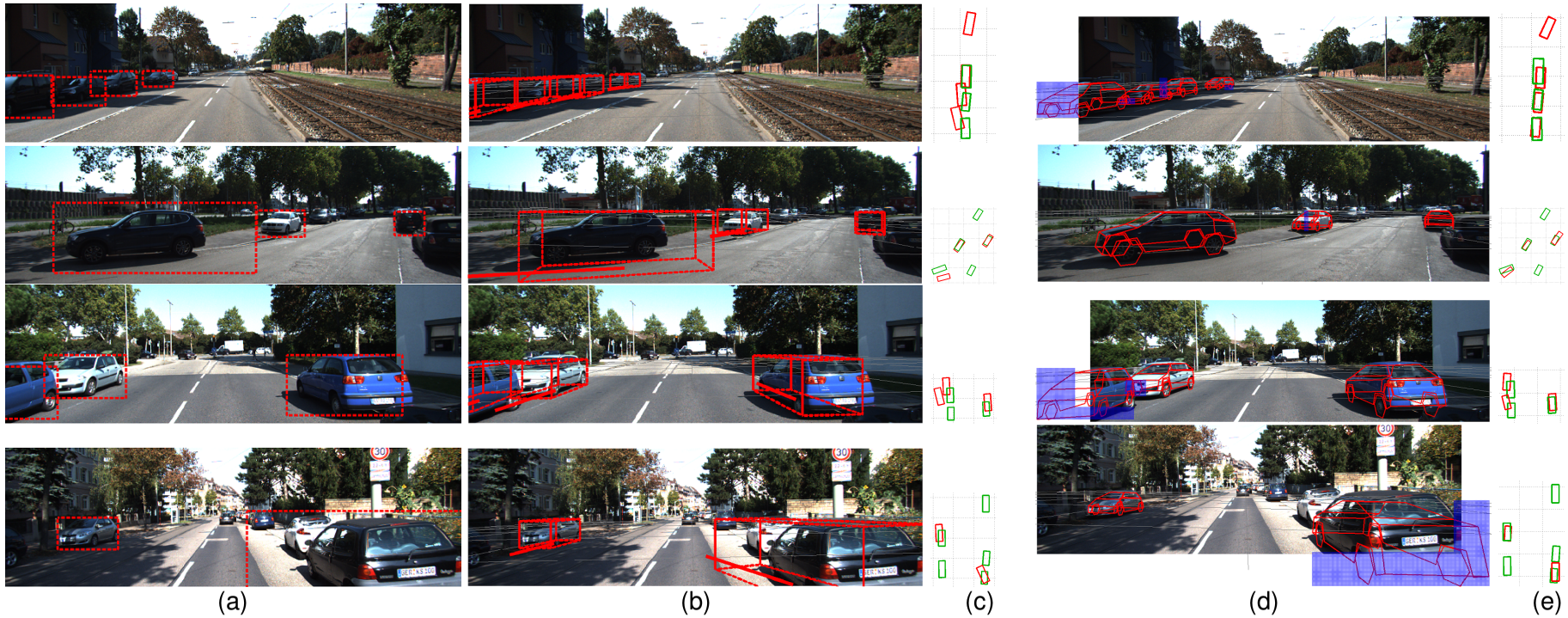


Figure 5.8: coarse+gp (a-c) vs fg+gp+do+so (d,e). (a) 2D bounding box detections, (b) coarse+gp based on (a), (c) bird's eye view of (b), (d) fg+gp+do+so shape model fits (blue: estimated occlusion masks), (e) bird's eye view of (d). Estimates in red, ground truth in green.

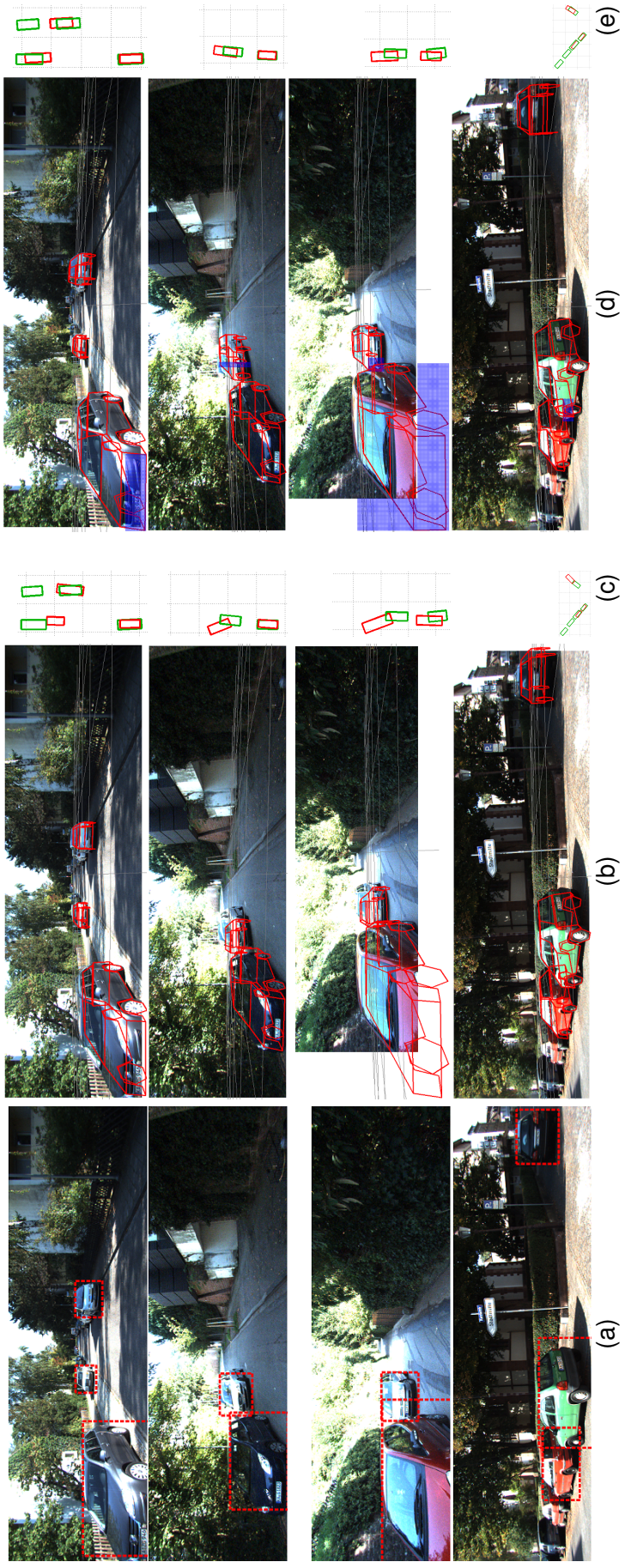


Figure 5.9: fg+gp (a-c) vs fg+gp+do+so (d,e). (a) 2D bounding box detections, (b) fg+gp based on (a), (c) bird's eye view of (b), (d) fg+gp+do+so shape model fits (blue: estimated occlusion masks), (e) bird's eye view of (d). Estimates in red, ground truth in green.

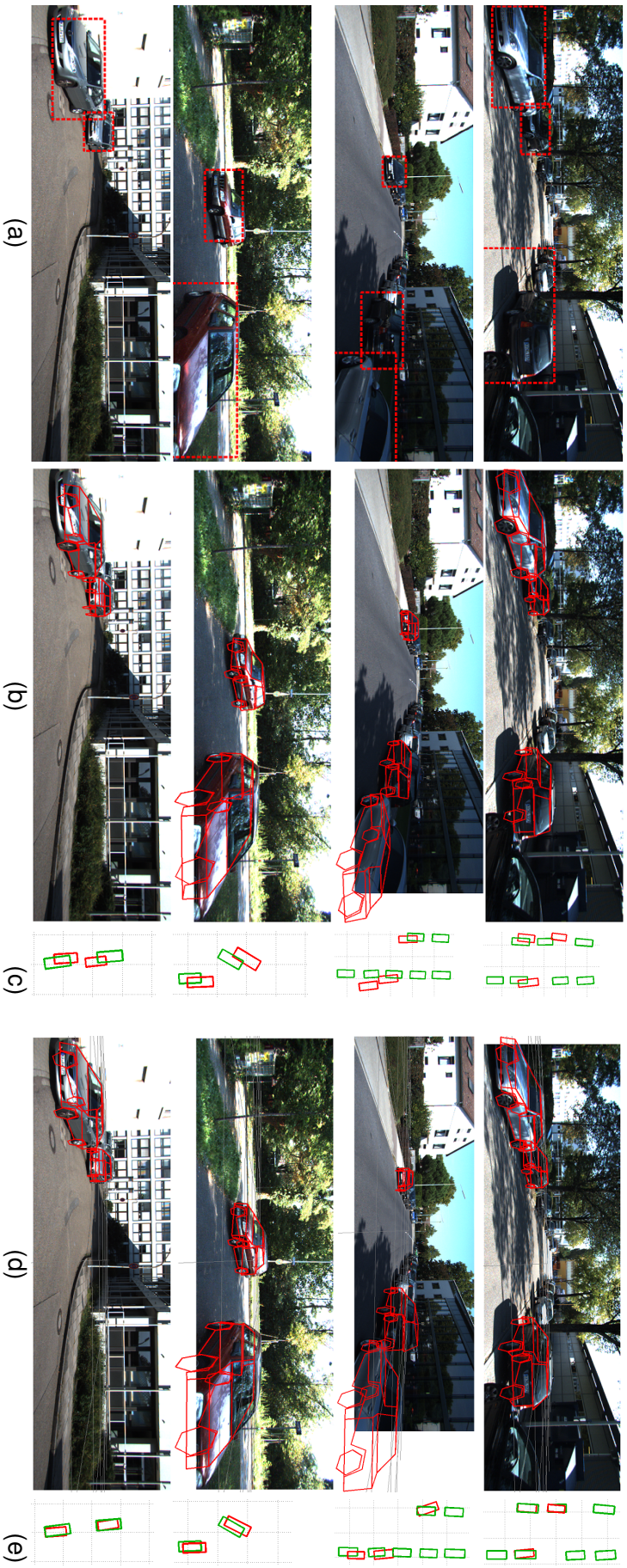


Figure 5.10: fg (a-c) vs fg+gp (d,e). (a) 2D bounding box detections, (b) fg based on (a), (c) bird's eye view of (b), (d) fg+gp shape model fits, (e) bird's eye view of (d). Estimates in red, ground truth in green.

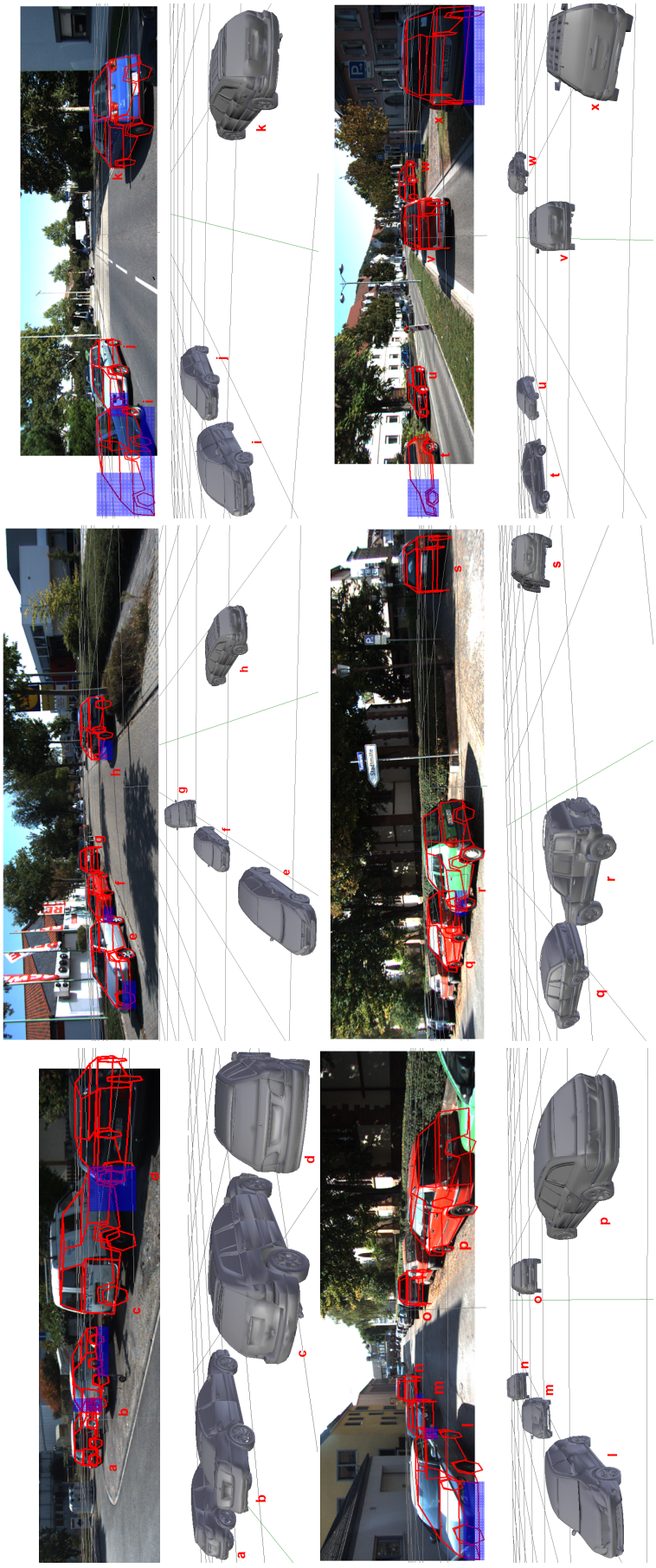


Figure 5.11: Example detections and corresponding 3D reconstructions.

Chapter 6

Conclusions and Outlook

Although 3D image interpretation had already been established as an engineering discipline in the middle of the 19th century studied by photogrammetrists (Schindler and Förstner, 2013), it was the Artificial Intelligence (AI) community which first started looking at automatic image interpretation in the 1960s. It took AI researchers many years before they came to appreciate the extremely challenging nature of the problem, with Marvin Minsky (one of the fathers of AI) famously claiming computer vision to be, "an undergraduate summer project" in 1966. Originating from AI, computer vision efforts in the early days were directed towards understanding the underlying 3D scene captured by an image. A number of ambitious attempts based on highly expressive models were made (Roberts, 1963; Binford, 1971; Nevatia and Binford, 1977; Marr and Nishihara, 1978; Brooks, 1981; Lowe, 1987) which proved far ahead of their time, given the extremely limited computational resources and unavailability of various low-level algorithms that only came into existence in the following decades. Consequently, computer vision research dispersed to work on these lower level problems ranging from shape-from-X to interest point detection and description, leveraging on insights from optics, graphics, statistics, operations research and differential analysis.

Over the last one and a half decade we have seen successes in a number of sub-problems: local features (Lowe, 1999; Tuytelaars and Gool, 2000; Belongie et al., 2000), image segmentation (Shi and Malik, 2000; Felzenszwalb and Huttenlocher, 2004), approximate inference (Isard and Blake, 1998; Murphy et al., 1999; Boykov et al., 2001), region labeling (He et al., 2004), multi-class discriminative classification (Breiman, 2001), and template matching (Viola and Jones, 2001; Dalal and Triggs, 2005). This has enabled researchers to revive the original goals that the community set out to achieve, that of detailed 3D scene understanding. A number of approaches have been proposed which coarsely reason about multiple scene components jointly in 3D, such that the components support each other in obtaining superior performance compared to independent detections (Geiger et al., 2011; Gupta et al., 2010; Hedau et al., 2010).

In this thesis we have attempted to go one step further and utilize finer-grained 3D models towards scene-level understanding. By revisiting a detailed 3D geometric model from the early days in the light of modern developments, we have demonstrated the usefulness of rich modeling even at the individual object level. Next, we have added the ability to reason about multiple object instances in a common 3D frame, adding a simple scene represen-

tation and modeling object-object interactions at a high resolution. We estimate a rich and surprisingly accurate 3D layout from a single view image, highlighting the potential of joint reasoning on detailed geometric models.

6.1 Discussion of contributions

We have explored both detaild 3D geometric modeling as well as applying such a model to the task of scene-level reasoning. The contributions resulting from the work thus also span both these domains.

6.1.1 Contributions to object class modeling

We explore a two layered approach to detailed 3D object class detection in Chapters 3 and 4. The purpose of the first layer is to provide full bounding box level detections together with coarse viewpoint, as well as being invariant to partial occlusions. The first layer accumulates votes from viewpoint dependent part *configurations*, and additionally leverages the individual *configuration* activations to predict local part locations. Our first layer outperforms the 2D detection results of *DPM* (Felzenszwalb et al., 2010) and original *poselets* (Bourdev and Malik, 2009), additionally providing coarse viewpoint estimates. We consider these results as a solid basis for subsequent 3D inference.

The second stage comprises of a 3D deformable wireframe model complementing early ideas in computer vision with modern techniques for robust model-to-image matching. This combination of 3D wireframes with discriminative local shape detectors allows for accurate estimation of object shape and continuous pose from single input images. In Chapter 3, we perform an extensive experimental study for 2D part-level localization, and continuous pose estimation, demonstrating accurate object geometry and viewpoint estimates on two challenging datasets for two object clases with very different geometry namely *cars* and *bicycles*. We show superior performance to Stark et al. (2010); Zia et al. (2011); Pepik et al. (2012b) on the task of continuous viewpoint estimation, demonstrating the value of reasoning about viewpoint in a continuous space. We further beat a naive baseline for 2D part localization (mean shape model in object bounding box) by a large margin. Intuitively speaking, these experiments confirm that, also for the purpose of machine vision, objects like cars or bicycles are not just 2D boxes or instances of a fixed template.

Finally, we demonstrate novel applications of such detailed shape estimation namely, ultra-wide baseline matching and fine-grained object categorization. In ultra-wide baseline matching, correspondences obtained from the detailed shape model outperform (by a large margin) interest point matching as well as matching independent part detections, showing that the object model indeed fulfills its job of providing a strong 3D prior model for object shape. We further beat not only our earlier work Zia et al. (2011) but also the latest works of Pepik et al. (2012b) and Pepik et al. (2012a), demonstrating the strength of fine-grained 3D representations. We demonstrate superior estimation of occlusion patterns as well as 2D part-level localization, as compared to an occlusion-agnostic baseline, highlighting the benefit of the simple, yet powerful occluder representation with masks. Overall,

the results support our hypothesis that detailed 3D geometry and occlusion modeling are beneficial even for independent object class recognition. We have made all our annotated training and test data as well as source code publicly available, which is already being used by other researchers.

6.1.2 Contributions to scene-level reasoning

Having demonstrated a powerful 3D geometric representation for object class instances, we move ahead to utilize this representation for jointly estimating shapes of multiple objects linked by a common scene geometry (ground plane) in Chapter 5. We demonstrate that a representation with occluder masks naturally includes both object-object occlusions as well as occlusions caused by unmodeled scene elements. It is, in fact, the detailed geometric hypotheses provided by our object model which enables us to choose a common ground plane touching the lower-most vertices of the wheels, as well as reason about object-object occlusions at the level of individual wireframe vertices. This is in contrast to full object bounding boxes which over-estimate the extent of the object.

We utilize a subset of the *KITTI* dataset (Geiger et al., 2012) for evaluating the contributions of different aspects of our scene model towards accuracies of: 3D object localization, 3D pose estimation, and 2D wireframe fitting and occlusion estimation. The evaluations are performed for three different settings corresponding to: the full dataset, only occluded objects, and only severely occluded objects. We demonstrate that both aspects of our model consistently improve 3D localization accuracy, with our full system giving the best performance across board. Similarly, we obtain the best performance for viewpoint estimation in the high accuracy regime with our full system. In fact, combining both aspects (occlusion modeling and common ground plane) gives us a striking 104–113% improvement in 3D localization as compared to just lifting 2D detections to 3D, as well as a reduction of the median orientation error from 13° to 5°. These improvements strongly support the case for 3D scene-level reasoning utilizing detailed models of object shape.

In terms of 2D localization accuracy the results are less clear cut. While true 3D reasoning does give minor improvements, the correlation is relatively weak. The lesson here is that the 3D reasoning should be evaluated in 3D space rather than in 2D projection. We will make the source code for scene-level reasoning available, too.

6.2 Technical evolution over the thesis

This thesis represents one coherent four year long project completed by sequentially reaching three milestones corresponding to the three core chapters (Chapters 3, 4, and 5). However, there has been a steady evolution in the sub-components of the overall system over the course of the project. Although the core chapters already specify these differences, we explicitly point them out in the following.

6.2.1 Initial detections

In Chapter 3, we initialize our inference using 2D detections from DPM-VOC-VP (Pepik et al., 2012b). This detector provides us with coarse 2D bounding box level hypotheses together with object pose discretized into eight bins. However, in order to cope with partial occlusions, we replace this detector with a bank of part-*configuration* detectors in Chapters 4 and 5. Besides obtaining full-object bounding box detections and coarse pose, this enables us to obtain additional evidence for local parts from the individual *configuration* activations in Chapter 4.

6.2.2 Changes in inference procedure

In Chapter 3, all variables in our search space are continuous and thus for each object hypothesis we are able to draw samples from a Gaussian proposal distribution centered on the previous value of the hypothesis. As we add an explicit occluder model in Chapter 4, the search space per object increases by a discrete variable (mask index) whose realizations have no obvious ordering. We achieve this ordering by defining a neighborhood between masks based on a rank order w.r.t. Hamming distance. Specifically we sort the set of masks w.r.t. the Hamming distance from the previous hypothesis, and then sample the offset in this ordering from a Gaussian.

We further need to modify the inference in Chapter 5, because our scene-level reasoning requires multiple objects to be modeled jointly, causing the number of search dimensions to increase many times. In order to deal with the resulting exponential increase in search space, we decouple the estimation of different objects by performing search for object shape and pose and search for occluder in alternating steps. This *block-coordinate descent* style inference additionally allows one to search exhaustively for the occluder mask indices instead of drawing random samples as in the earlier stages of the work.

6.2.3 From pseudo-3D to true 3D

In Chapters 3 and 4, we fit projections of our detailed 3D object model inside 2D bounding boxes predicted by the coarse detection layer. However, in Chapter 5 we lift our representation to metric 3D, to enable modeling of interactions between multiple object hypotheses in a common 3D frame. This is achieved by training the geometric model on 3D CAD data scaled according to real-world dimensions rather than normalized dimensions (which are enough for fitting in 2D scale), and then performing a 2D-to-3D lifting in an intermediate layer (between first and second layer) based on a grid-search for 3D object pose and ground plane parameters.

6.2.4 Part location prediction from first layer

In Chapter 4, we utilize the activations of part *configuration* detectors to predict part locations. The predictions are incorporated as an additional term in the objective function, based on Gaussians learnt on 2D distributions of relative part locations from the training data. We demonstrated superior 2D part localization results using this feature of our model as compared to when it was disabled. The performance boost was moderate, but

clearly greater for heavily occluded cases.

In Chapter 5 when evaluating over the KITTI dataset (Geiger et al., 2012) we observed a slight loss in 2D localization performance over the *full dataset*, when using this additional cue (but no difference in 3D performance). We explain this discrepancy as follows: in the case of occlusions where some part patches might be partially occluded causing inaccurate responses from the part detectors for those windows, incorporating larger context helps improve the part location estimates. However for fully visible objects, such predictions can actually mislead the wireframe model causing a worse fit. This happens because the predictions for part locations from larger partial object (part *configuration*) detectors are less precise than those from the part detectors themselves. Thus, over the *full dataset* of Chapter 5, which has twice as many fully visible cars as occluded ones, we get an overall slight loss of performance. Since, the focus there is on 3D scene analysis, we disable and omit mention of this cue altogether.

6.3 Limitations of our approach

This thesis has succeeded in making relevant contributions to detailed 3D geometric modeling and scene-level reasoning, but of course there is room for improvement. The following point out some limitations of the current work.

Improvement in 2D detections. One key limitation of the approach is that it does not lead to improvement in terms of 2D detections, in fact, we consistenly lose a couple of detections in all our experiments. Intuitively, finer-grained reasoning about object parts and contextual reasoning should lead to overall improved bounding box level detections, however surprisingly neither this thesis nor other 3D geometric modeling approaches have as yet (Pepik et al., 2012b; Xiang and Savarese, 2013) succeeded in beating the performance of relatively coarse models.

Object classes considered. Although the approach can, in principle, model any rigid object class with a well-defined topology, we experimentally evaluate only two classes with very different geometry: *cars* and *bicycles* in Chapter 3 and only *cars* in Chapters 4 and 5. One reason is the unavailability of suitable datasets, specially a test set with scenes comprising of multiple objects with 3D annotations. However, experimenting with more object classes can yield valuable insights in developing scene-level reasoning for more general scene types.

Non-rigid objects. Our object model itself is limited to modeling topologically consistent rigid object classes and cannot handle articulated objects (*e.g.* , humans), objects with fairly weak global shape (*e.g.* cats and trees), and functional object categories (*e.g.* chairs). We expect a more sophisticated 3D scene-level reasoning system to contain multiple types of object models which can accomodate such non-rigid object classes.

Single supporting plane. Our single ground plane assumption while being very helpful for the scenes where it physically holds, can be restrictive at times. We go around this limitation in our current implementation, by switching off the ground plane for cases

where our intermediate layer cannot find a single plane hypotheses over which all the detected objects lie within reasonable tolerance. However, a more sophisticated approach which allows multiple planes or more general surfaces perhaps aided by object detections as well as low-level cues, could be applicable to a broader range of scenes.

Early commitment to hypotheses. Currently we choose all object detections scoring above a threshold in our first layer, and refine this set of detections in the second layer. A more robust approach would be to not commit to a fixed set of hypotheses in the first layer, and allow for addition and deletion of hypotheses based on the 3D reasoning.

Processing speed. Another issue with the current implementation are the long times need by the inference scheme to converge, which for some images with multiple object instances can take as much as 30-40 minutes per image. Unfortunately this is a well-known problem with simulation-based approaches, *e.g.* even the latest 3D face model fitting algorithms (Schönborn et al., 2013) require similar amounts of processing time.

6.4 Outlook

In this section, we mention potential solutions for overcoming the limitations discussed above. Further we discuss proposals for future research both w.r.t. technical features and broader implications of successful fine-grained 3D modeling.

6.4.1 Detailed 3D object modeling

Since we want to get accurate part-level fits as opposed to just 2D bounding boxes, even in cluttered scenes and under significant lighting variations, a relevant direction for further investigation to improve the object models would be to incorporate cues which can pull the wireframe model towards object boundaries. While this concerns making the appearance model more sophisticated, another important future direction is to explore detailed 3D models which may be relevant for more non-rigid, articulated, and functional object classes.

Perceptual grouping. A powerful cue for separating an object instance from background clutter useful for some object classes such as cars is segmenting out the object from the background. This can be based on an actual foreground-background segmentation algorithm (Parkhi et al., 2011) or leveraging on self-similarity (Deselaers and Ferrari, 2010). Such cues can be utilized best in a model-driven framework, *e.g.* to neglect transparent windows *vs.* car body for the segmentation. Interaction of top-down reasoning with bottom-up processes of segmenting out object regions may be boot-strapped from an initial detection and then iteratively refined.

Boundary and edge features. Explicitly fitting to edge pixels on the object boundaries as well as inside the object can also serve as a valuable cue for precise estimation of the deformable wireframe (Schindler and Suter, 2008; Zia et al., 2009; Payet and Todorovic, 2011). The challenge is to reject spurious edges due to background clutter

and to optimally weigh between utilization of complementary features (Stark et al., 2009).

Part regressors and structured output learning. We can achieve better 3D wireframe fits and potentially speed up the inference by making better use of the training data. One potential approach is to replace part appearance classifiers with regressors which can predict the pixel offset to correct part location given a nearby patch (Cristinacce and Cootes, 2007). An orthogonal approach is to use structured output learning approaches (Tsochantaridis et al., 2004; Blaschko and Lampert, 2008; Pepik et al., 2012a), learning object localization, viewpoint, and even 3D object shape and occlusion parameters jointly. Both these directions not only learn a classifier to distinguish between perfectly localized object or part windows *vs.* background windows, but also utilize partially overlapping windows on training data to improve detection accuracy.

General object geometry. As mentioned in Section 6.3, one important limitation of our object representation is that it can only model rigid object classes with a well-defined topology. An important research direction would be to enhance the representation to also handle articulations (Sigal and Black, 2006) and possibly non-rigid but topologically consistent object classes (Cashman and Fitzgibbon, 2013): estimating pose and shape for the visible portion of an object instance, as well as a distribution on plausible pose and shape hypotheses for the occluded portion. Another important question is how point-based shape analysis can be extended to learn prior shape models for object classes which do not possess a topologically consistent membership, *e.g.* buildings as viewed in aerial photographs. Efforts in this direction could enable the approach developed in this thesis to be applied to visual analysis problems in other domains such as remote sensing and medical image interpretation.

6.4.2 Scene-level reasoning

For scene-level reasoning, we propose to incorporate further priors that describe high-level interactions among object instances, to model more scene elements, and to make the inference more robust and efficient. We mention some ideas along these lines in the following:

High-level object interaction priors. Although we inject some prior knowledge in the form of object-object occlusions and a common ground plane assumption into our scene model, there are many other high-level constraints which could be leveraged on. For example, in our current setting of street scene analysis, we can further enforce long-range regularities, since cars parked at the roadside as well as those on the road are usually parallel. Similarly in the case of indoor scenes, we can bias our inference towards commonly occurring configurations of objects, such as computers over office desks, or chairs under dining tables.

Modeling other scene elements. Currently, we only detect object instances and reason about their interactions. However we can additionally model other scene elements, such as building façades, poles, trees, street surface, and incorporate these detections into our scene-level reasoning. This would constrain the search space, as well as provide stronger 3D priors on object locations.

Multiple supporting planes. As a single ground plane is currently a limitation of our system, an obvious extension is to pursue modeling more general terrains while retaining the benefits in terms of 3D localization accuracy. The first obvious attempt to achieve this would be to use more than one planar segment, incorporating low-level image features apart from object-level votes and trying a statistical model selection scheme to find the number of planes appropriate for a given scene. Similarly for indoor scenes, fitting planar segments to supporting surfaces such as table-tops could yield improved estimates. We foresee the inference algorithm generating these sophisticated scene hypotheses to follow an iterative approach, refining object hypotheses lying on these planes alternately with the supporting planes themselves (Hedau et al., 2009) - thus improving both types of estimates.

Delayed commitment to hypotheses. As mentioned in Section 6.3, our approach currently commits itself to a fixed set of object hypotheses from the first layer (2D detections) based on a threshold on detection scores. A more principled approach would be to give more 2D hypotheses a chance to be evaluated in 3D space, and decide on the suitability of hypotheses after detailed reasoning: whether to keep or delete the hypothesis. A sampling framework which provides such capabilities is the Reversible Jump Markov Chain Monte Carlo (RJMCMC) approach (Green, 1995): which allows adding, switching, and deleting hypotheses and has been successfully applied in a number of recent scene-level reasoning works (Wojek et al., 2013; Xiang and Savarese, 2013; Del Pero et al., 2013).

Processing speed. The large amount of time required to process each test image would be a major hurdle in widespread adoption of detailed 3D scene understanding, as proposed here. While during this thesis we focus on evaluating the potential of such fine-grained reasoning, there are clear hints which can be followed vis-à-vis improvement in efficiency. Our inference approach maintains a set of scene-level hypotheses called *particles* which are refined independent of each other over a number of iterations. The particle updates can be trivially parallelized given a suitable computational platform. Further, our inference procedure computes sampling variances on the fly (Leordeanu and Hebert, 2008), which can cause the sampler to draw the same samples many times. Thus implementing an efficient data structure where we store the objective values for each already visited point in the search space, to avoid re-evaluating the objective function for the same hypothesis later, can yield further speed ups. Apart from these immediate possibilities, application-oriented optimization can also be applied, *e.g.* if the approach is applied to a tracking scenario (in video), the number of particles maybe be reduced, and stronger priors placed on detections and viewpoints based on temporal regularities to achieve massive per-frame speed ups.

6.4.3 The big picture

The broader question that we investigate in this thesis is whether fine-grained 3D modeling really helps scene-level reasoning. We rely on relatively large amounts of manual annotation for definition and annotation of object parts. With massive amounts of

visual data being freely available such as 3D CAD models (on free databases like Google 3D Warehouse), there is a potential to avoid the need for even minor offline human interaction, and scale the system to far more complex and diverse scenes. Secondly, such fine-grained models do not have to be confined to the domain of single image understanding, and there is a great potential for such approaches in settings where multiple views of the scene are available or where the scene is dynamic. We mention these research directions in the following:

Scalability issues The approach presented in this thesis, as well as similar recent work (Del Pero et al., 2013; Xiang and Savarese, 2013) requires significant amounts of manual annotation effort (at the level of individual parts) and currently only handles a few object classes. Thus an important future direction is to investigate approaches that reduce annotation effort and another is to reduce the computational cost for large scale object detection. Possibilities for reducing annotation effort include either simplifying registered 3D CAD models automatically downloaded from internet by making approaches such as Zia et al. (2009) more efficient, or automatically inferring parts and correspondences across CAD exemplars based on 3D geometry (Shalom et al., 2008). For large scale object detection, we need to share both the appearance and geometric representation among many object classes. The issue of appearance sharing has been treated in many works, usually accomplished by either sharing of visual words among object classes (Krempp et al., 2002; Torralba et al., 2004; Bart and Ullman, 2005; Opelt and Pinz, 2006; Stark et al., 2009), by representing objects using a shared hierarchy of parts (Zhu et al., 2010; Salakhutdinov et al., 2011), or by representing HOG-style filters trained separately for different object classes as a sparse combination of basis filters (Song et al., 2012). Thus the bigger challenge is to share the global geometry representation, since the part detections themselves are noisy, and a prior geometry model acts as a strong regularizer in such approaches. One idea is to investigate some form of hierarchical geometric representation which follows a coarse-to-fine approach to estimate geometry shape.

A string of recent successes in learning complex concepts come from the revival of “deep” learning approaches (Socher et al., 2012; Sermanet et al., 2013; Mnih et al., 2013). One direction for future research would thus be to utilize the huge amounts of visual data available online, such as video repositories or 3D animated movies, and attempt to learn explicit 3D models of scene layouts and objects (perhaps segmented and reconstructed on the basis of motion) in a deep learning framework, perhaps bootstrapped with some knowledge of physics and most commonly occurring objects.

Multiple views and depth cameras. We exclusively consider the single view setting in this thesis. However in many application such as robotics, augmented reality, and surveillance, multiple views of the scene are available. While multiple views from a monocular camera, or a view from a stereo or depth camera already provide more information about the scene, w.r.t. to 3D geometry and otherwise occluded regions, there still exists a huge potential for detailed semantic modeling to resolve correspondences, reconstruct textureless and highly specular surfaces, and cluster together regions of the scene as belonging to the same object. Researchers have begun exploring such opportunities (Schindler and Bauer, 2003; Gee et al., 2008; Bao and Savarese, 2011;

Silberman et al., 2012; Salas-Moreno et al., 2013; Fioraio and Stefano, 2013; Dame et al., 2013; Satkin and Hebert, 2013), however much remains to be done in terms of coherently combining 3D reconstruction with semantic recognition. One immediate reward of incorporating our style of detailed semantic modeling into a *visual SLAM* pipeline is a qualitatively rich estimate of the scene from the very first frame that sees an unexplored portion of the environment. This single-frame estimate can then be quantitatively improved as the environment is explored and new frame are seen, which can be very beneficial for applications such as augmented reality or in service or disaster response robots. Another interesting extension of such research would be in the domain now called "life-long learning", where semantic concepts, such as planar approximations, detailed 3D geometric models, as well as physics modeling, could be combined to learn new object instances and classes on the fly as well as different attributes of objects, like affordances.

Richer motion models. While we have restricted our discussion to static scenes, detailed dynamic scene understanding requires not only modeling the geometry of objects at a high-resolution, but also their motion. Researchers have recently started reasoning about the movements and dynamic interactions of objects. This work includes detailed temporal modeling efforts such as "social" motion models (Pellegrini et al., 2009; Luber et al., 2010; Baumgartner et al., 2013): modeling the interaction between moving objects in time domain, *e.g.* a group of people moving together or a person dragging a shopping cart. Similarly, recent dynamic scene understanding approaches (Ess et al., 2009; Wojek et al., 2013; Geiger et al., 2011) leverage different motion models for different object classes, *e.g.* constant velocity model for pedestrians, and mechanically constrained motion model for cars. However, the underlying object shape models remain rather coarse, which hinder a more deeper analysis of the motion. Detailed 3D geometric models can enable fine-grained dynamic scene understanding by providing precise viewpoint and high-resolution part estimates, which would feed into richer motion models. These detections would allow to better estimate and constrain 3D motion: for rigid objects like the motion of a shopping cart, and for articulated objects such as the possible motions of car doors, a bicycle, or a human body. Thus an interesting direction for future work is to investigate richer motion models that can best utilize the increased expressiveness provided by detailed object models.

Appendix A

Bibliography

S. Agarwal, D. Roth, Learning a Sparse Representation for Object Detection, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002

Y. Amit, D. Geman, Shape Quantization and Recognition with Randomized Trees. Neural Computation **9**(7), 1545–1588 (1997)

M. Andriluka, S. Roth, B. Schiele, Pictorial structures revisited: People detection and articulated pose estimation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009

M. Andriluka, S. Roth, B. Schiele, Discriminative Appearance Models for Pictorial Structures. International Journal on Computer Vision (IJCV) **99**(3), 259–280 (2011)

M. Arie-Nachimson, R. Basri, Constructing Implicit 3D Shape Models for Pose Estimation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009

S.Y. Bao, S. Savarese, Semantic Structure from Motion, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011

O. Barinova, V. Lempitsky, E. Tretyak, P. Kohli, Geometric image parsing in man-made environments, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010

E. Bart, S. Ullman, Corss-generalization: Learning novel classes from a single example by feature replacement, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005

T. Baumgartner, D. Mitzel, B. Leibe, "Tracking People and Their Objects", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

S. Belongie, J. Malik, J. Puzicha, Shape Context: A New Descriptor for Shape Matching and Object Recognition, in *Advances in Neural Information Processing Systems (NIPS)*, 2000

T.O. Binford, Visual Perception by Computer, in *Proceedings of the IEEE Conference on Systems and Control*, 1971

C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, USA, 2007)

M.B. Blaschko, C.H. Lampert, Learning to Localize Objects with Structured Output Regression, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008

L. Bourdev, J. Malik, Poselets: Body part detectors trained using 3D human pose annotations, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009

S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, UK, 2004)

Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **23**(11), 1222–1239 (2001)

L. Breiman, *Classification And Regression Trees* (Chapman and Hall, London, UK, 1984)

L. Breiman, Bagging predictors. Machine Learning **24**(2), 123–140 (1996)

L. Breiman, Random forests. Machine Learning **45**(1), 5–32 (2001)

R.A. Brooks, Symbolic reasoning among 3-d models and 2-d images. Artificial Intelligence **17**(1), 285–348 (1981)

M. Brown, D. Lowe, Recognising panoramas, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003

C.G. Broyden, J.E. Dennis Jr., J.J. More, On the Local and Superlinear Convergence of Quasi-Newton Methods. IMA Journal of Applied Mathematics **12**(3), 223–245 (1973)

T. Cashman, A. Fitzgibbon, What shape are dolphins? Building 3D morphable models from 2D images. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **35**(1), 232–244 (2013)

D.M. Chen, S.S. Tsai, R. Vedantham, R. Grzeszczuk, B. Girod, Streaming Mobile Augmented Reality on Mobile Phone, in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009

Y. Chen, T.-K. Kim, R. Cipolla, Inferring 3D Shapes and Deformations from Single Views, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010

W. Choi, Y.-W. Chao, C. Pantofaru, S. Savarese, Understanding Indoor Scenes Using 3D Geometric Phrases, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active Shape Models - Their Training and Application. Computer Vision and Image Understanding (CVIU) **61**(1), 38–59 (1995)

N. Cornelis, B. Leibe, K. Cornelis, L. Van Gool, 3D urban scene modeling integrating recognition and reconstruction. International Journal on Computer Vision (IJCV) **78**(2-3), 121–141 (2008)

D. Cristinacce, T. Cootes, Boosted Regression Active Shape Models, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2007

G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in *Proceedings of the Workshop on Statistical Learning in Computer Vision*, 2004

N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005

A. Dame, V.A. Prisacariu, C.Y. Ren, I. Reid, Dense Reconstruction Using 3D Object Shape Priors, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **29**(6), 1052–1067 (2007)

L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, K. Barnard, Bayesian geometric modeling of indoor scenes, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012

L. Del Pero, J. Bowdish, B. Kermgard, E. Hartley, K. Barnard, Understanding Bayesian rooms using composite 3D object models, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

T. Deselaers, V. Ferrari, Global and efficient self-similarity for object classification and detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010

M. Enzweiler, A. Eigenstetter, B. Schiele, D.M. Gavrila, Multi-Cue Pedestrian Classification with Partial Occlusion Handling, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010

A. Ess, B. Leibe, K. Schindler, L.V. Gool., Robust multi-person tracking from a mobile platform. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **31**(10), 1831–1846 (2009)

M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge. International Journal on Computer Vision (IJCV) **88**(2), 303–338 (2010)

R. Farrell, O. Oza, N. Zhang, V.I. Morariu, T. Darrell, L.S. Davis, Birdlets: Subordinate Categorization Using Volumetric Primitives and Pose-Normalized Appearance, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011

P.F. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **32**(9) (2010)

P.F. Felzenszwalb, D.P. Huttenlocher, Efficient Graph-Based Image Segmentation. International Journal on Computer Vision (IJCV) **59**(2), 167–181 (2004)

R. Fergus, P. Perona, A. Zisserman, Object Class Recognition by Unsupervised Scale-Invariant Learning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003

N. Fioraio, L.D. Stefano, Joint detection, tracking, and mapping by semantic bundle-adjustment, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

M. Fischler, R. Elschlager, The representation and matching of pictorial structures. IEEE Transactions of Computer **22**(1), 67–92 (1973)

R. Fransens, C. Strecha, L.V. Gool, A Mean Field EM-algorithm for Coherent Occlusion Handling in MAP-Estimation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006

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

Y. Freund, R.E. Shapire, Experiments with a new boosting algorithm, in *Proceedings of the International Conference on Machine Learning (ICML)*, 1996

J. Gall, V. Lempitsky, Class-Specific Hough Forests for Object Detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009

T. Gao, B. Packer, D. Koller, A Segmentation-aware Object Detection Model with Occlusion Handling, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011

A.P. Gee, D. Chekhlov, A. Calway, W. Mayol-Cuevas, Discovering Higher Level Structure in Visual SLAM. IEEE Transactions on Robotics **24**(5), 980–990 (2008)

A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012

A. Geiger, C. Wojek, R. Urtasun, Joint 3D Estimation of Objects and Scene Layout, in *Advances in Neural Information Processing Systems (NIPS)*, 2011

R.B. Girshick, P.F. Felzenszwalb, D. McAllester, Object detection with grammar models, in *Advances in Neural Information Processing Systems (NIPS)*, 2011

D. Glasner, M. Galun, S. Alpert, R. Basri, G. Shakhnarovich, Viewpoint-aware object detection and pose estimation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011

F. Glover, Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research **13**(5), 533–549 (1986)

P.J. Green, Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. Biometrika **82**(4), 711–732 (1995)

C. Gu, X. Ren, Discriminative mixture-of-templates for viewpoint classification, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010

A. Gupta, A.A. Efros, M. Hebert, Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010

M. Haag, H.-H. Nagel, Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. International Journal on Computer Vision (IJCV) **35**(3), 295–319 (1999)

R.I. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edn. (Cambridge University Press, Cambridge, UK, 2004)

X. He, R.S. Zemel, M.A. Carreira-Perpinan, Multiscale Conditional Random Field for Image Labeling, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004

V. Hedau, D. Hoiem, D.A. Forsyth, Recovering the Spatial Layout of Cluttered Rooms, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009

V. Hedau, D. Hoiem, D.A. Forsyth, Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010

M. Hejrati, D. Ramanan, Analyzing 3D Objects in Cluttered Images, in *Advances in Neural Information Processing Systems (NIPS)*, 2012

A.v.d. Hengel, A. Dick, T. Thormählen, B. Ward, P.H.S. Torr, VideoTrace: Rapid interactive scene modeling from video. ACM Transactions on Graphics **26**(3) (2007)

D. Hoiem, S. Savarese, *Representations and Techniques for 3D Object Recognition and Scene Interpretation* (Morgan and Claypool Publishers, California, USA, 2011)

D. Hoiem, A. Efros, M. Hebert, Putting objects in perspective. International Journal on Computer Vision (IJCV) **80**(1), 3–15 (2008)

D. Hoiem, A.A. Efros, M. Hebert, Automatic Photo Pop-up, in *Proceedings of the International Conference and Exhibition on Computer Graphics and Interactive Techniques (ACM SIGGRAPH)*, 2005

J.H. Holland, *Hidden Order: How Adaptation Builds Complexity* (Addison-Wesley, Redwood City, California, USA, 1975)

M. Isard, A. Blake, CONDENSATION – conditional density propagation for visual tracking. International Journal on Computer Vision (IJCV) **1**(29), 5–28 (1998)

T. Kanade, A Theory of Origami World. Artificial Intelligence (1980)

G. Klein, D. Murray, Parallel Tracking And Mapping for Small AR Workspaces, in *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007

D. Koller, K. Daniilidis, H.H. Nagel, Model-based object tracking in monocular image sequences of road traffic scenes. International Journal on Computer Vision (IJCV) **10**(3), 257–281 (1993)

S. Krempp, D. Geman, Y. Amit, Sequential learning of reusable parts for object detection. Technical Report (2002)

S. Kwak, W. Nam, B. Han, J.H. Han, Learning occlusion with likelihoods for visual tracking, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011

S. Lazebnik, C. Schmid, J. Ponce, Sparse texture representation using affine-invariant neighborhoods, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003

S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006

B. Leibe, A. Leonardis, B. Schiele, An implicit shape model for combined object categorization and segmentation. Toward Category-Level Object Recognition (2006)

C. Leistner, Semi-Supervised Ensemble Methods for Computer Vision, PhD thesis, TU Graz, 2010

M. Leordeanu, M. Hebert, Smoothing-based Optimization, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008

M.J. Leotta, J.L. Mundy, Vehicle surveillance with a generic, adaptive, 3d vehicle model. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **33**(7) (2011)

V. Lepetit, P. Fua, Keypoint recognition using randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **28**(9), 1465–1479 (2006)

Y. Li, L. Gu, T. Kanade, Robustly aligning a shape model and its application to car alignment of unknown pose. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **33**(9) (2011)

J. Liebelt, C. Schmid, Multi-View Object Class Detection with a 3D Geometric Model, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010

J. Liebelt, C. Schmid, K. Schertler, Viewpoint independent object class detection using 3D Feature Maps, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008

R.J. Lopez-Sastre, T. Tuytelaars, S. Savarese, Deformable Part Models Revisited: A Performance Evaluation for Object Category Pose Estimation, in *Proceedings of the IEEE Workshop on Challenges and Opportunities in Robot Perception (ICCV WS CORP)*, 2011

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

D.G. Lowe, Object recognition from local scale-invariant features, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1999

D.G. Lowe, Distinctive image features from scale invariant keypoints. International Journal on Computer Vision (IJCV) **2**(60), 91–110 (2004)

M. Luber, J.A. Stork, G.D. Tipaldi, K.O. Arras, "People Tracking with Human Motion Predictions from Social Forces", in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2010

J. Malik, Interpreting Line Drawings of Curved Objects. International Journal on Computer Vision (IJCV) **1**(1), 73–103 (1987)

D. Marr, H.K. Nishihara, Representation and recognition of the spatial organization of three-dimensional shapes. Proceedings of the Royal Society of London. Series B, Biological Sciences **200**(1140), 269–294 (1978)

D. Meger, C. Wojek, B. Schiele, J.J. Little, Explicit occlusion reasoning for 3d object detection, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2011

B.H. Menze, B.M. Kelm, D.N. Splitthoff, U. Koethe, F.A. Hamprecht, On oblique random forests, in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2011

N. Metropolis, S. Ulam, The Monte Carlo method. Journal of the American Statistical Association **44**(247), 335–341 (1949)

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equations of State Calculations by Fast Computing Machines. Journal of Chemical Physics **21**(6), 1087–1092 (1953)

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

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with Deep Reinforcement Learning. pre-print (2013)

K.P. Murphy, Y. Weiss, M.I. Jordan, Loopy belief propagation for approximate inference: an empirical study, in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999

R. Nevatia, T.O. Binford, Description and recognition of curved objects. Artificial Intelligence (AI), 77–98 (1977)

R.A. Newcombe, S.J. Lovegrove, A.J. Davison, DTAM: Dense Tracking And Mapping in real-time, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011

M. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in *Proceedings of the Indian Conference on Computer Vision, Graphics, and Image Processing (ICVGIP)*, 2008

A. Opelt, A. Pinz, Incremental learning of object detectors using a visual shape alphabet, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006

M. Ozuysal, V. Lepetit, P. Fua, Pose estimation for category specific multiview object localization, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009

O.M. Parkhi, A. Vedaldi, C.V. Jawahar, A. Zisserman, The truth about cats and dogs, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011

N. Payet, S. Todorovic, From Contours to 3D Object Detection and Pose Estimation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011

S. Pellegrini, A. Ess, K. Schindler, L. van Gool, "You'll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009

A. Pentland, Perceptual organization and representation of natural form. Artificial Intelligence (AI) **28**(3), 293–331 (1986)

B. Pepik, M. Stark, P. Gehler, B. Schiele, Occlusion Patterns for Object Class Detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

B. Pepik, P. Gehler, M. Stark, B. Schiele, 3DDPM - 3D Deformable Part Models, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012a

B. Pepik, M. Stark, P. Gehler, B. Schiele, Teaching 3D Geometry to Deformable Part Models, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012b

J.R. Quinlan, Induction of Decision Trees. Machine Learning **1**(1), 81–106 (1986)

I. Rechenberg, Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, PhD thesis, TU Berlin, 1971

L.G. Roberts, Machine Perception of Three-Dimensional Solids, PhD thesis, MIT, 1963

R. Salakhutdinov, A. Torralba, J.B. Tennenbaum, Learning to share visual appearance for multiclass object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011

R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H.J. Kelly, A.J. Davison, SLAM++: Simultaneous localization and mapping at the level of objects, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

S. Satkin, M. Hebert, 3DNN:Viewpoint Invariant 3D Geometry Matching for Scene Understanding, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013

S. Savarese, L. Fei-Fei, 3D generic object categorization, localization and pose estimation., in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007

K. Schindler, J. Bauer, A model-based method for building reconstruction, in *Proceedings of the IEEE Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, 2003

K. Schindler, W. Förstner, *Photogrammetry* (chapter in Encyclopedia of Computer Vision, Katsushi Ikeuchi (EiC), Springer Reference, 2013)

K. Schindler, D. Suter, Object detection by global contour shape. Pattern Recognition **41**(12), 3736–3748 (2008)

H. Schneiderman, T. Kanade, A Statistical Method for 3D Object Detection Applied to Faces and Cars, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000

S. Schönborn, A. Forster, B. Egger, T. Vetter, A Monte Carlo Strategy to Integrate Detection and Model-Based Face Analysis, in *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2013

P. Sermanet, D. Eigen, X. Zhang, M. M., R. Fergus, Y. LeCun, OverFeat: Integrated Recognition, Localization and Detection using Convolution Networks. pre-print (2013)

S. Shalom, L. Shapira, A. Shamir, D. Cohen-Or, Part analogies in sets of objects, in *Proceedings of Eurographics Symposium on 3D Object Retrieval*, 2008

D.F. Shanno, Conditioning of quasi-Newton methods for function minimization. Mathematics of Computation **24**(111), 647–656 (1970)

Shape Context, *Shape Context — Wikipedia, The Free Encyclopedia*, 2013. [Online; accessed 7-January-2014]. `http://en.wikipedia.org/wiki/Shape_context`

J. Shi, J. Malik, Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **22**(8), 888–905 (2000)

L. Sigal, M.J. Black, Predicting 3D People from 2D Pictures, in *Proceedings of the Conference on Articulated Motion and Deformable Objects (AMDO)*, 2006

N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor Segmentation and Support Inference from RGBD Images, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012

J. Sivic, A. Zisserman, "Video Google: A text retrieval approach to object matching in videos", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003

R. Socher, B. Huval, B. Bhat, C.D. Manning, A.Y. Ng, "Convolutional-Recursive Deep Learning for 3D Object Classification", in *Advances in Neural Information Processing Systems (NIPS)*, 2012

H.O. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, T. Darrell, Sparselet Models for Efficient Multiclass Object Detection, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012

M. Stark, M. Goesele, B. Schiele, A Shape-Based Object Class Model for Knowledge Transfer, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009

M. Stark, M. Goesele, B. Schiele, Back to the Future: Learning Shape Models from 3D CAD Data, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2010

H. Su, M. Sun, L. Fei-Fei, S. Savarese, Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories., in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009

G.D. Sullivan, A.D. Worrall, J.M. Ferryman, Visual Object Recognition Using Deformable Models of Vehicles, in *Proceedings of the IEEE Workshop on Context-Based Vision*, 1995

M. Sun, B.X. Xu, G. Bradski, S. Savarese, Depth-Encoded Hough Voting for joint object detection and shape recovery, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010

S. Tang, M. Andriluka, B. Schiele, Detection and Tracking of Occluded People, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2012

A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, L. Van Gool, Towards Multi-View Object Class Detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006

A. Torralba, K.P. Murphy, W.T. Freeman, "Sharing Features: efficient boosting procedures for multiclass object detection", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004

L. Torresani, A. Hertzmann, C. Bregler, Learning non-rigid 3d shape from 2d motion, in *Advances in Neural Information Processing Systems (NIPS)*, 2003

I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, Support vector machine learning for interdependent and structured output spaces, in *Proceedings of the International Conference on Machine Learning (ICML)*, 2004

T. Tuytelaars, L.V. Gool, Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2000

T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey. Foundations and Trends in Computer Graphics and Vision (2008)

A. Vedaldi, B. Fulkerson, *VLFeat: An Open and Portable Library of Computer Vision Algorithms*, 2008

A. Vedaldi, A. Zisserman, Structured output regression for detection with partial truncation, in *Advances in Neural Information Processing Systems (NIPS)*, 2009

M. Villamizar, H. Grabner, J. Andrade-Cetto, A. Sanfeliu, L.V. Gool, F. Moreno-Noguer, Efficient 3D Object Detection using Multiple Pose-Specific Classifiers, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2011

P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001

D. Wagner, A. Mulloni, D. Schmalstieg, Real-Time Detection and Tracking for Augmented Reality on Mobile Phone. IEEE Transactions on Visualization and Computer Graphics **16**(3), 355–368 (2010)

H. Wang, S. Gould, D. Koller, Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010

X. Wang, T.X. Han, S. Yan, An HOG-LBP human detector with partial occlusion handling, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009

C. Wojek, S. Roth, K. Schindler, B. Schiele, Monocular 3D Scene Modeling and Inference: Understanding Multi-Object Traffic Scenes, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010

C. Wojek, S. Walk, S. Roth, K. Schindler, B. Schiele, Monocular visual scene understanding: understanding multi-object traffic scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **35**(4), 882–897 (2013)

Y. Xiang, S. Savarese, Object Detection by 3D Aspectlets and Occlusion Reasoning, in *Proceedings of the IEEE International Workshop on 3d Representation and Recognition (ICCV WS 3dRR)*, 2013

Y. Xiang, S. Savarese, Estimating the Aspect Layout of Object Categories, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012

P. Yan, S.M. Khan, M. Shah, 3D Model based Object Class Detection in An Arbitrary View, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007

B. Yao, A. Khosla, L. Fei-Fei, Combining Randomization and Discrimination for Fine-Grained Image Categorization, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011

Y. Zhao, S.-C. Zhu, Scene Parsing by Integrating Function, Geometry and Appearance Models, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

L.L. Zhu, Y. Chen, A. Torralba, W. Freeman, A. Yuille, Part and appearance sharing: Recursive Compositional Models for Multi-View Multi-Object Detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010

M.Z. Zia, U. Klank, M. Beetz, Acquisition of a dense 3D model database for robotic vision, in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, 2009

M.Z. Zia, M. Stark, K. Schindler, Explicit Occlusion Modeling for 3D Object Class Representations, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

M.Z. Zia, M. Stark, K. Schindler, Are Cars Just 3D Boxes? – Jointly Estimating the 3D Shape of Multiple Objects, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014a

M.Z. Zia, M. Stark, K. Schindler, Towards scene understanding with detailed 3D object representations. Submitted to journal (2014b)

M.Z. Zia, M. Stark, K. Schindler, B. Schiele, Revisiting 3D Geometric Models for Accurate Object Shape and Pose, in *Proceedings of the IEEE International Workshop on 3d Representation and Recognition (ICCV WS 3dRR)*, 2011

M.Z. Zia, M. Stark, B. Schiele, K. Schindler, Detailed 3d representations for object recognition and modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **35**(11), 2608–2623 (2013)

Appendix B

Publication List

[1] M.Z. Zia, M. Stark, K. Schindler, Towards Scene Understanding with Detailed 3D Object Representations, Submitted to journal, 2014

[2] M.Z. Zia, M. Stark, K. Schindler, Are Cars Just 3D Boxes? – Jointly Estimating the 3D Shape of Multiple Objects, *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014 (Accepted)

[3] M.Z. Zia, M. Stark, B. Schiele, K. Schindler, Detailed 3D representations for object recognition and modeling, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), November 2013

[4] M.Z. Zia, M. Stark, K. Schindler, Explicit Occlusion Modeling for 3D Object Class Representations, *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

[5] M.Z. Zia, M. Stark, K. Schindler, Towards Scene Understanding with Detailed 3D Object Representations, *Abstracts of Scene Understanding Workshop (SUNw)*, 2013

[6] M.Z. Zia, M. Stark, B. Schiele, K. Schindler, Revisiting 3D geometric models for accurate object shape and pose, *in Proceedings of the 3rd International IEEE Workshop on 3D Representation and Recognition (3dRR)*, 2011 (Best Paper Award)

[7] M.Z. Zia, U. Klank, and M. Beetz, Acquisition of Dense 3D Model Database for Robotic Vision, *in Proceedings of the International Conference on Advanced Robotics (ICAR)*, 2009

[8] U. Klank, M.Z. Zia, and M. Beetz, 3D Model Selection from an Internet Database for Robotic Vision, *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009

Appendix C

Curriculum Vitae

Education

Oct 2010 - Dec 2013	**Swiss Federal Institute of Technology, Zurich** PhD Student
Oct 2007 - Sep 2009	**Munich University of Technology** MSc, Electrical Communication Engineering
Jan 2003 - Dec 2006	**NED University of Engineering and Technology** BEng, Electronic Engineering

Experience

Jan 2014 -	**Imperial College London** Post-Doctoral Research Associate
Oct 2010 - Dec 2013	**Swiss Federal Institute of Technology, Zurich** Research and Teaching Assistant, Web-Master
May 2013 - Aug 2013	**Qualcomm Research, Vienna** Visiting Researcher
Oct 2009 - Sep 2010	**Darmstadt University of Technology** Research Assistant
Nov 2007 - Sep 2009	**Munich University of Technology** Student Researcher
Jul 2008 - Sep 2008	**Siemens AG, Munich** Engineering Intern

Jan 2007 - Sep 2007	**NED University of Engineering and Technology, Karachi** Teaching Assistant
Apr 2006 - Dec 2006	**Pakistan Space and Upper Atmosphere Research Commission** Engineering Intern
Dec 2005 - Jan 2006	**Siemens Pakistan Engineering, Karachi** Engineering Intern