TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Künstliche Intelligenz, Lehrstuhl für Bildverarbeitung

HYPOTHESIS GENERATION FOR OBJECT POSE ESTIMATION FROM LOCAL SAMPLING TO GLOBAL REASONING

Frank Michel

DISSERTATION

TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik Institut für Künstliche Intelligenz, Lehrstuhl für Bildverarbeitung

HYPOTHESIS GENERATION FOR OBJECT POSE ESTIMATION FROM LOCAL SAMPLING TO GLOBAL REASONING

Frank Michel
Born on: 12th January 1982 in Eisenach

DISSERTATION

to achieve the academic degree

DOKTOR RERUM NATURALIUM (DR. RER. NAT.)

First referee
Prof. PhD. Carsten Rother

Second referee
Prof. Dr. Carsten Steger

Advisor
Prof. Dr. Stefan Gumhold

Supervisor
Prof. PhD. Carsten Rother

Submitted on: 27th November 2017
Defended on: 18th January 2018

ABSTRACT

Pose estimation has been studied since the early days of computer vision. The task of object pose estimation is to determine the transformation that maps an object from it's inherent coordinate system into the camera-centric coordinate system. This transformation describes the translation of the object relative to the camera and the orientation of the object in three dimensional space. The knowledge of an object's pose is a key ingredient in many application scenarios like robotic grasping, augmented reality, autonomous navigation and surveillance. A general estimation pipeline consists of the following four steps: extraction of distinctive points, creation of a hypotheses pool, hypothesis verification and, finally, the hypotheses refinement. In this work, we focus on the hypothesis generation process. We show that it is beneficial to utilize geometric knowledge in this process.

We address the problem of hypotheses generation of articulated objects. Instead of considering each object part individually we model the object as a kinematic chain. This enables us to use the inner-part relationships when sampling pose hypotheses. Thereby we only need K correspondences for objects consisting of K parts. We show that applying geometric knowledge about part relationships improves estimation accuracy under severe self-occlusion and low quality correspondence predictions. In an extension we employ global reasoning within the hypotheses generation process instead of sampling 6D pose hypotheses locally. We therefore formulate a Conditional-Random-Field operating on the image as a whole inferring those pixels that are consistent with the 6D pose. Within the CRF we use a strong geometric check that is able to assess the quality of correspondence pairs. We show that our global geometric check improves the accuracy of pose estimation under heavy occlusion.

Statement of authorship

I hereby certify that I have authored this Dissertation entitled *Hypothesis Generation for Object Pose Estimation From local sampling to global reasoning* independently and without undue assistance from third parties. No other than the resources and references indicated in this thesis have been used. I have marked both literal and accordingly adopted quotations as such. There were no additional persons involved in the intellectual preparation of the present thesis. I am aware that violations of this declaration may lead to subsequent withdrawal of the degree.

Dresden, 27th November 2017

Frank Michel

ACKNOWLEDGEMENT

This is the beginning of end of a tremendous adventure that started six years ago and I would like to take the opportunity to thank the people that accompanied and supported me during this time. A true adventure it was that took me through the depths of computer vision and machine learning. Luckily, I wasn't fighting on my own. In Eric and Alex I found the two best colleagues I could have ever asked for. I deeply enjoyed the time we spent together, even the sleepless nights before the deadlines. I truly believe that our success was only possible because of the environment we created as a team which also made us grow both on an academical as well as on a personal level. I am also thankful for all the conversations and discussions I had with André, Joachim, Sebastian, and Sasha.

I want to thank my supervisor Carsten Rother for his advice, guidance, and for giving me a direction to follow, as well as my advisor Stefan Gumhold for his feedback and for letting me choose my own path of exploration.

I am sincerely thankful for the love and support I have received from my family and friends. I am grateful to my parents, who have provided me unconditional support through my studies, and to Maritta for her support and for her excitement for what I am doing. I also want to thank Basti for being a true friend for as long as I can think.

I want to thank Tina for being part of my life for so man years. This thesis would have not been possible without your everlasting support and I am glad that you are proud of me and excited about what I am doing even after all the pain that this project has caused. I am glad to have you by my side. Finally, I want to thank Vico and Nevi for showing me what is really important in life.

CONTENTS

Abstract 4

1. Introduction 14
   - 1.1. Pose Estimation Task . . . 16
     - 1.1.1. Challenges . . . 18
     - 1.1.2. Related Tasks . . . 19
     - 1.1.3. Applications . . . 23
   - 1.2. Overview . . . 26
     - 1.2.1. Contributions . . . 27
   - 1.3. Related Work . . . 27
     - 1.3.1. Exhaustive Search . . . 28
     - 1.3.2. Voting-based Approaches . . . 30
     - 1.3.3. Sampling-based Approaches . . . 31
   - 1.4. Outline . . . 33

2. 6D Pose Estimation 34
   - 2.1. Introduction . . . 34
   - 2.2. Background . . . 35
     - 2.2.1. Decision Trees . . . 35
     - 2.2.2. Object Coordinate Regression . . . 36
     - 2.2.3. Hypothesis Scoring . . . 38
   - 2.3. Method . . . 39
     - 2.3.1. Training Data Generation . . . 39
     - 2.3.2. Pose Estimation by Random Sampling . . . 40
   - 2.4. Experiments . . . 41
   - 2.5. Summary . . . 42

3. Articulated Pose Estimation 44
   - 3.1. Introduction . . . 44
   - 3.2. Related Work . . . 46
   - 3.3. Method . . . 47
     - 3.3.1. The Articulated Pose Estimation Task . . . 47
     - 3.3.2. Object Coordinate Regression . . . 48
     - 3.3.3. Hypothesis Generation . . . 48

3.3.4. Energy Optimization . . . 50
3.4. Experiments . . . 50
3.4.1. Dataset . . . 50
3.4.2. Setup . . . 51
3.4.3. Results . . . 53
3.5. Summary . . . 55

4. Global Hypothesis Generation 58
4.1. Introduction . . . 58
4.2. Related Work . . . 61
4.3. Background - Graphical Models . . . 62
4.4. Method . . . 63
4.4.1. Global Reasoning . . . 63
4.4.2. Method - Graphical Model . . . 64
4.4.3. Energy Minimization . . . 65
4.4.4. Pose Estimation as Energy Minimization . . . 65
4.4.5. Stage One: Problem Size Reduction . . . 67
4.4.6. Stage Two: Generation of Solution Candidates . . . 67
4.4.7. On Optimality of Subproblem Solutions for Binary Energy Minimization . . . 70
4.4.8. Obtaining Candidates for Partial Optimal Labeling . . . 71
4.4.9. Refinement and Hypothesis Scoring . . . 72
4.5. Experiments . . . 72
4.5.1. Dataset . . . 73
4.5.2. Results . . . 73
4.6. Summary . . . 74

5. Discussion 76
5.1. Accuracy . . . 76
5.2. Scalability . . . 77
5.3. Global Hypothesis Generation as an End-to-End Pipeline . . . 78
5.4. 6D Pose Challenge . . . 79

6. Conclusion 82

A. Appendix 84
A.1. Abbreviations . . . 84
A.2. Datasets . . . 85
A.2.1. Occlusion Datasets . . . 85
A.2.2. Articulated Objects Dataset (Our) . . . 86
A.3. Derivation for the Estimation of Articulation Parameters. . . . 88

List of Figures 92

LIST OF PULISHED PAPERS

We will discuss the following three papers in detail within this thesis.

1. Pose Estimation of Kinematic Chain Instances via Object Coordinate Regression [78]
   Frank Michel, Alexander Krull, Eric Brachmann, Michael Ying Yang, Stefan Gumhold, Carsten Rother
   BMVC 2015 (Oral Presentation)

2. Introducing LiDAR Point Cloud-based Object Classification for Safer Apron Operations [79]
   Johannes Mund, Frank Michel (shared first), Franziska Dieke-Meier, Hartmut Fricke, Lothar Meyer, Carsten Rother
   ESAVS 2016 (Oral Presentation)

3. Global Hypothesis Generation for 6D Object Pose Estimation [77]
   Frank Michel, Alexander Kirillov, Eric Brachmann, Alexander Krull, Stefan Gumhold, Bogdan Savchynskyy, Carsten Rother
   CVPR 2017 (Spotlight Presentation)

We also contributed to the following publications addressing various tasks related to 6D pose estimation. We will however not discuss them in this thesis.

4. Learning 6D Object Pose Estimation using 3D Object Coordinates [8]
   Eric Brachmann, Alexander Krull, Frank Michel, Jamie Shotton, Stefan Gumhold, Carsten Rother
   ECCV 2014 (Poster Presentation)

5. 6-DOF Model Based Tracking via Object Coordinate Regression [66]
   Alexander Krull, Frank Michel, Eric Brachmann, Stephan Ihrke, Stefan Gumhold, Carsten Rother
   ACCV 2014 (Oral Presentation and Honorable Mention Demo Award)

6. Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images [64]
   Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, Carsten Rother
   ICCV 2015 (Poster Presentation)

7. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a single RGB Image [10]
   Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, Carsten Rother
   CVPR 2016 (Poster Presentation)

8. DSAC - Differentiable RANSAC for Camera Localization [9]
   Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, Carsten Rother
   CVPR 2017 (Oral Presentation)

9. PoseAgent: Budget-Constrained 6D Object Pose Estimation via Reinforcement Learning [65]
   Alexander Krull, Eric Brachmann, Sebastian Nowozin, Frank Michel, Jamie Shotton, Carsten Rother
   CVPR 2017 (Poster Presentation)

1. INTRODUCTION

Contents

1.1. Pose Estimation Task 16
1.1.1. Challenges 18
1.1.2. Related Tasks 19
1.1.3. Applications 23
1.2. Overview 26
1.2.1. Contributions 27
1.3. Related Work 27
1.3.1. Exhaustive Search 28
1.3.2. Voting-based Approaches 30
1.3.3. Sampling-based Approaches 31
1.4. Outline 33

The development of digital electronic computers which began in the middle of the 20th century changed people's life tremendously. Early computers occupied rooms or even buildings. They were designed and operated by only a few experts to solve highly specialized tasks. The progress in electrical engineering enabled pocket-sized computers that are ubiquitous today. Watches, glasses or telephones are all quipped with computers and their ability to efficiently compute and automate tasks improved people's living conditions. This becomes evident in technologies like smart homes, computer aided surgery and the internet.

While the size of computers decreased during the development process their computational power increased. This facilitates solving more complex tasks and processing large amounts of data. With the progress in the field of artificial intelligence we are at the verge of a new level of automation. Autonomous driving cars, automated warehouses, parcel delivering drones and supermarkets with neither cashiers nor checkout stations are not far from becoming reality. Those technologies will again heavily influence and simplify peoples daily life.

While robots were heavy machines only working in an isolated and controlled environment (e.g. a welding robot in a car factory, only a few years ago) they are now operating in many domestic environments where they are cleaning the floor or cutting the lawn in the garden. Working in such uncontrolled environments requires the robot

to perceive and interpret their surroundings in order to navigate and interact automatically. Such tasks are of high complexity and humans mostly use the large bandwidth of their visual system to solve them. While perceptual psychologists investigated the human visual system for decades, a full understanding of the complex vision process still remains elusive [116]. Similar to the human vision system cameras are able to capture rich visual representations of the real world. The progress in the field of electrical engineering enabled cameras to be readily available and small, e.g. modern mobile telephones are all equipped with at least one camera.

Computer vision attempts to equip machines with the ability to see and extract information from visual data automatically. A broad range to tasks are addressed in the computer vision domain, e.g. object recognition, segmentation, and tracking. In this work, we are approaching the problem of accurate object pose estimation. Knowing the pose of an object is a prerequisite for solving tasks like obstacle avoidance in autonomous navigation or grasping in robotic object manipulations. As humans, we start training for this task at the age of 12 weeks [120] and we constantly enhance our dexterities. We develop an abstract knowledge enabling us to recognize objects under different lighting conditions, in the presence of clutter and occlusion, and even objects that change their shape, e.g. deformable objects. Employing this knowledge makes pose estimation an easy task for humans. Creating this knowledge and modeling the complexity of the visual world is challenging which makes the task of pose estimation difficult for machines.

Early computer vision approaches started with the similar task of reconstructing the 3D structure of objects from 2D images in order to understand the depicted scene. Marr [73] proposed a general bottom-up approach to solve this task. The structure of the approach was adopted from the human visual system where edges and boundaries are composed to an initial sketch that builds the foundation for the generation of a 2.5D representation which is furthermore used to create a 3D model of the scene. Roberts [96] also attempted the task of scene understanding by identifying lines on texture-less polyhedral objects and reconstructing their 3D shapes using a library of polyhedral block components. The main idea of both approaches was to find correspondences between positions on the object and positions within the image.

Research moved from using lines as correspondence features to edges [14] and later to corners [34] which enabled detection of more general objects. With the introduction of SIFT (scale invariant feature transform) by Lowe [71] feature descriptors gained importance. They were used to describe distinctive object features while being robust to translational and rotational movement and to changing object scales. The SIFT descriptor was utilized in various tasks like the creation of 3D shapes from multiple images taken under different viewpoints [13], object pose estimation [33], and simultaneous localization and mapping (SLAM) [53]. While the SIFT descriptor focused on single image positions the histogram of gradients (HOG) descriptor introduced by Dalal and Triggs [18] captured features on an object level. The early approaches to computer vision were mainly hand-crafted methods, meaning that they contain parameters that needed to be tuned for different scenarios. This changed when machine learning techniques started pushing into the domain of computer vision in the early 2000s.

With more high quality training data becoming available, methods learning significant features from data gained popularity. The algorithm that builds the foundation of

the Microsoft Kinect body pose estimation proposed by Taylor et al. [117] employed the random forest framework to extract expressive features from training data. Those features were used to establish correspondences between positions in the image and positions on the human body. Deep neural networks, a machine learning concept first introduced by Ivanenhko [46, 47] is now applied to many computer vision problems. In recent years convolutional neural networks (CNN) gained influence and achieved state-of-the-art results on many computer vision tasks, e.g. 2D object detection [93] and semantic segmentation [127]. They were also applied to the domain of 3D object detection [24] and pose estimation [125].

Hypothesize and test, a strategy first popularized by Forsyth and Ponce [30], is a well-established procedure in the computer vision domain. It divides the task into two steps. The first step uses a subset of the available information to create a hypothesis. This is followed by a verification process using all available information to determine how well the hypothesis aligns with the data and selecting the hypothesis as the final output that achieved the highest score. The strategy is employed in the popular RANSAC algorithm developed by Fischler and Bolles [28] in 1981 and it is still utilized today.

Decomposing the task into elements enables the application of what Roberts [96] called the "laws of nature", meaning the geometric knowledge that is available about the task. Most recently Brachmann et al. [9] showed that it is beneficial to use this task knowledge within the camera re-localization process. They outperformed the method of Kendall et al. [58], which omits this knowledge and learns a CNN to directly regress the camera position from the input image.

In this work we are adopting the hypothesize and test strategy and apply it to the task of object pose estimation. We are in particular focusing on the hypothesis generation process and show that the application of geometric task knowledge does not only lead to improved hypotheses, it also reduces the required effort.

In the remainder of this chapter we will first introduce the task at hand formally (Section 1.1), discuss challenges (Section 1.1.1) and variants (Section 1.1.2) of the pose estimation task and show potential application scenarios (Section 1.1.3).

1.1. POSE ESTIMATION TASK

There is a extensive variety of object types appearing in the real world. Different object types reach from rigid objects (e.g. a tea cup) over articulated objects (e.g. a laptop) to deformable objects (e.g. a sponge). This variety also reflects in the methods addressing the task of object pose estimation. These methods are typically differentiated into instance-based and class-based approaches. Object instances are explicitly defined by their shape and texture which distinguishes them from all other object instances. Object classes, in contrast, are generic and contain multiple similar objects. Passenger aircraft is an example of such an object class which is defined by the object shape through the aircraft fuselage and the wings. An Airbus 319-100 is a specific instance of this object class. In this work we address the task of 6D object instance pose estimation of rigid and deformable, in particular, articulated objects.

The tasks of object recognition and object detection are related to the task of pose

estimation but they differ with respect to the location information they provide about the object. Object recognition determines whether a given image contains the object of interest but does not provide any information about the object localization. Object detection on the opposite detects the presence of the object in the image and provides the location of the object, e.g. as a bounding box for the case of 2D detection and as a bounding volume when approaching 3D detection. While the object position is accurately determined by detection methods the orientation of the object is not or only roughly estimated. Object pose estimation determines both the 3D position and the 3D orientation of an object. While methods working on pose estimation of object classes often only provide a discretized and therefore coarse estimate of the object orientation we aim for a continuous and accurate pose estimate which is essential for applications like object grasping and augmented reality.

FORMAL DEFINITION

Given an RGB-D image I we want to determine the pose H of an object that is described by the object position and object orientation relative to the camera. The pose is defined as a rigid body transformation, where the object orientation is determined by a 3×3 rotation matrix $\mathbb{R}$ describing a rotation around the object center, and a 3×1 translation vector t describing the position of the object in the camera coordinate system. This requires a digital 3D model of the object, which we assume to be given. Such a model can either be generated synthetically using computer aided design or by using 3D reconstruction methods. Furthermore, we do not aim for object detection, but assume that only one instance of the object is present in the image.

HYPOTHESIZE AND TEST STRATEGY

There exists a great variety of different approaches addressing the task of pose estimation. Many of them follow the abstract concept of hypothesize and test [30]. We will introduce this concept based on the task of pose estimation and divide it into three steps. The first step establishes correspondences between the object and the image, meaning that a correspondence relates a position in the image to a position on the object. There are many techniques solving the correspondence problem ranging from hand-crafted features like SIFT [71] or FPFH [98] to machine learning techniques like random forests or CNNs. Independent of the technique employed, correspondences can be erroneous and prone to contain incorrect matches, termed outliers. Using those outliers during the pose estimation process will most likely lead to an incorrect pose prediction.

The goal of the second step, the hypothesis generation, is to find at least one outlier free correspondence set. Sampling based methods, e.g. RANSAC [28], use a minimal set of correspondences required for the task to reduce the risk of including an outlier. Voting based methods accumulate many correspondences supporting the same hypothesis. Since the focus of this thesis lies on the second step of the pose estimation pipeline we will give a broad overview of related methods in Chapter 1.3. The third and final step determines how well the hypotheses are explained by the image data and selects the best scoring hypothesis as the final output. This often contains a

refinement process, e.g. the iterative closest point algorithm (ICP), to align the initial pose hypothesis to the data.

INPUT SENSOR DATA

RGB cameras are predominantly used in computer vision domain. They capture rich visual information, are cheap to produce, and come in sizes that allows their installation in devices of every day use like mobile phones. RGB sensors are designed to capture visible light. This causes them to be sensitive to changing lighting conditions since the object appearance is substantially altered if the lighting conditions change. RGB features also rely on objects to contain texture leading to poor results for textureless objects whose appearance is often only defined by their contours. Furthermore, determining the object depth using RGB is difficult since small distance changes are not reflected in the captured image, due to the camera resolution or other pixel quantization effects.

Sensors measuring distances from the camera to positions in the scene, so called depth sensors, were predominately used in the robotics community. Laser-based depth sensors scan the environment by sending out laser beams and measuring the runtime until the beams returns. This information is utilized to generate unstructured point cloud data represented as 3D coordinates. Depth sensors employing the structure-from-light concept are based on the stereoscopic vision principle. Those sensors either use two cameras or a camera-projector-setup to determine the distance of an image position by utilizing the triangulation concept. Depth sensors are more expensive and, although their size decreased, considerably larger than RGB sensors. This changed with the introduction of the Mircosoft Kinect sensor, a small and cheap device providing aligned RGB and depth (RGB-D) data, which popularized the utilization of depth data in the computer vision domain. Depth sensors do not rely on capturing the visual light which makes their measurements invariant to lighting changes. They do however fail to provide reliable measurements on highly reflective materials and on sharp edges. There is large variety of other sensor technologies available, e.g. radar, ultrasonic or thermal imaging sensors. We omit them in this work since they were not considered for accurate object pose estimation.

The methods presented in this thesis rely on depth and RGB sensor data. We did, however, contribute to the works of Brachmann et al. [10, 9] considering the tasks of object pose estimation and camera re-localization using RGB data only.

1.1.1. CHALLENGES

The task of object pose estimation has been investigated for decades and is not solved yet. We will introduce the major challenges arising with the task of instance pose estimation.

LIGHTING CHANGES

Lighting has a strong influence on how objects are perceived. While bright light casts hard shadows introducing artificial color edges on the object, colored light effects the

wavelength of light that is reflected by the object. Humans are able to recognize objects even if their appearance is heavily influenced by the lighting conditions because they develop a general understanding of how lighting changes the appearance of the object. It is difficult to model this knowledge and computer vision techniques relying purely on object textures struggle in providing reliable results under varying lighting conditions. Depth sensors, however, are not influenced by the described phenomena and are therefore invariant to changing lighting conditions.

OCCLUSION AND CLUTTER

The difficulties arising by cluttered environments and object occlusions are severe. Objects positioned in the proximity distort the object boundaries in both RGB and depth data. This especially impedes the performance of features relying on object contours. The pose estimation process can furthermore be distracted if objects in the vicinity show similarities of color and shape.

Occlusions introduce an additional challenge for pose estimation methods. Often objects only occupy a small fraction of the pixels in the image. Occlusions will not only lead to a reduction of the visible surface of the object, they can also cover distinctive features like a handle of a cup rendering the unambiguous pose estimation process impossible.

AMBIGUITIES IN OBJECT SHAPE AND TEXTURE

Many objects in industrial environments lack in distinctive texture disabling methods that rely purely on RGB features like SIFT [71]. The growing interest in warehouse automation by the industry motivated researcher to address pose estimation of objects showing textural ambiguities. This lead to the creation of several methods and data sets approaching pose estimation of texture-less objects [37, 41, 39, 94].

Pose estimation is ambiguous if the object furthermore lacks in distinctive shape. Such object symmetries distract the pose estimation process since multiple pose outputs are possible for one given input image. A texture-less bowl is an example for such ambiguities. A metric addressing those issues was proposed by Hodan et al. [40].

REFLECTIONS

Reflective surfaces are also affected by the above described lighting changes. Highlights created by bright light distract RGB as well as depth sensors. Reflections of the environment can furthermore overlay onto the object texture causing a drastic change in appearance. Those effects are dependent on the viewpoint on the object and therefore challenging to model. Both RGB and depth sensors suffer from this effect and fail to provide reliable information.

1.1.2. RELATED TASKS

The task of object pose estimation can be extended and variegated in many ways. In the following, we will shortly discuss the variants that are closely related to the

Figure 1.1.: **Pose Estimation Challenges.** From left to right: object showing ambiguities in both shape and texture, partial object occlusion and background clutter, surrounding environment reflected on the object shape, different lighting conditions changing the appearance of the object.

methods proposed in this thesis.

POSE ESTIMATION OF ARTICULATED OBJECTS

Man-made as well as biological objects, e.g. humans or animals, are often assemblies of multiple rigid parts connected through links. Furniture like a cabinet containing drawers and doors or a laptop are examples of rigid articulated objects. One could consider parts of an articulated object as single instances and estimate their poses individually, but this will most likely result in low quality pose estimates. This is caused by the inherent nature of articulated objects where parts can draw large occlusions onto other parts of the object, an effect that we refer to as self occlusion. Using the underlying structure of the articulated object within the pose estimation process resolves issues caused by self occlusion. An example would be a closed laptop where only a fraction of the laptop body is visible. Incorporating the knowledge of the object structure enables pose estimation methods to concentrate their effort for the occluded part to the close vicinity of the visible object part, in this case the laptop lid yields indications for the position of the laptop body. In Chapter 3 we show how the relationships between parts can be used within the hypothesis generation process to improve pose estimation of articulated objects.

MULTI-OBJECT AND MULTI-INSTANCE POSE ESTIMATION

The assumption that the object of interest is present and only occurs once does not always hold. A task where a robot should automatically assemble multiple work pieces, e.g. composing gears into a clock mechanism, would violate the aforementioned assumptions. The robot needs to detect the object in order to estimate the pose and should not be distracted by an object instance that occurs multiple times. In this scenario the algorithm needs to take this into account by estimating the pose of each object instance present in the scene. We contributed to the method of Brachmann et al. [10] where object detection is encapsulated within the hypothesis generation pro-

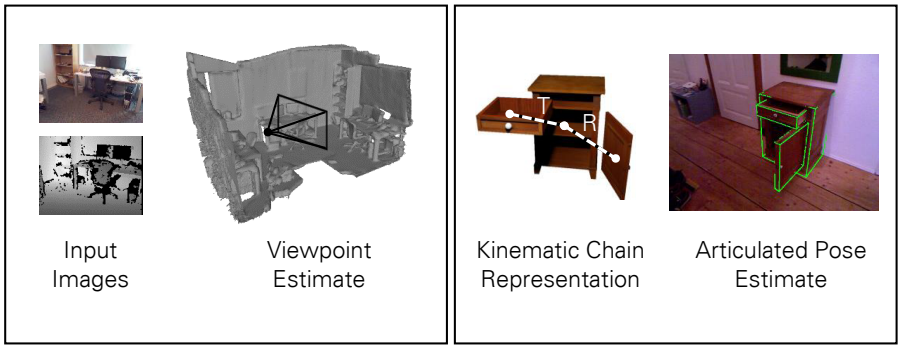


Figure 1.2.: **Camera Re-localization and Articulated Objects. Left:** Given an image, camera re-localization determines the position of the camera within a known environment where the image was acquired. RGB-D input images are shown on the left, the camera pose is visualized by a black camera frustum. **Right:** Articulated objects are assemblies of rigid parts connected by joints. The cabinet object shown on the left is connected to the drawer via a prismatic and to the door via a revolute joint. This underlying structure can be utilized for pose estimation of articulated objects. Pose estimates of the individual parts are shown as green bounding boxes.

cess by concentrating only on high scoring objects and rejecting object hypotheses with limited support. Although the methods proposed in this thesis can be extended to the multi-instance case, we did not consider this scenario in our work.

POSE TRACKING

For stationary objects performing pose estimation on a single image is sufficient. A moving robot can use other sensor modalities, like odometry, to measure its motion and recalculate the position of the object without estimating the pose from image data. If the object itself is moving within the scene the object needs to be tracked purely based on image data. Object tracking can be conducted by processing each time frame individually and ignoring the knowledge about previous estimates, an approach termed "tracking by detection". On the contrary, filtering approaches like the the Kalman filter [51] and the particle filter introduced by del Moral [19] employ estimates of previous time steps within a probabilistic model to track the object. This does not only improve the robustness with respect to occlusions it also reduces the computational effort and therefore increases the speed. We will not address the problem of pose tracking. However, Krull et al. [66] proposed a method where object coordinate regression, a foundation of our work, is used within the particle filter framework to increase robustness towards occlusion and fast object movement.

Figure 1.3.: **Multiple Objects and Multiple Instances.** We show different setups for the task of instance pose estimation ranging from the simple single object - single instance to the more challenging multi object.

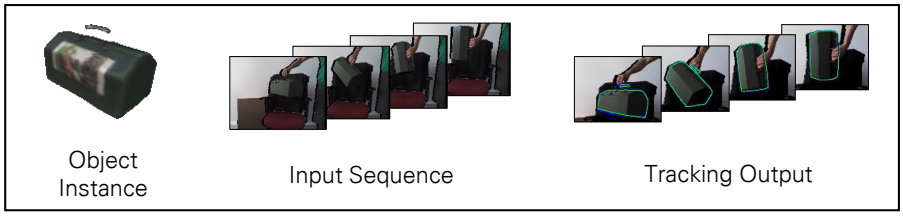


Figure 1.4.: **Pose Tracking.** Instead of estimating the object pose for a single image, object tracking is conducted on a series of consecutive images in which the object of interest is moving. The 6D object pose estimates are depicted as green bounding boxes.

CAMERA RE-LOCALIZATION

The previously discussed variants considered small objects occurring in a larger scene. For the task of camera localization or re-localization the scene is considered to be the object. Given an RGB or RGB-D image the camera position capturing in the scene should be determined. This task is similar to scenario described by Engelson and McDermott [25] as the kidnapped robot problem, where the robots is moved to an arbitrary location which forces the robot to re-estimate its position purely based on image data. While the camera re-localization task faces the same challenges as object pose estimation their influence on the process is different. The effect of occlusions is usually smaller since the correspondence estimation is not restricted to a small part of the image. However, ambiguities in shape and texture are more severe since scenes might contain repetitive structures (e.g. stairs in an stairway) or regions with neither color nor depth feature (e.g. monochrome walls) leading to ambiguous correspondences. Camera re-localization of outdoor scenes (e.g. operating on a cityscape level) does also pose it own difficulties.

1.1.3. APPLICATIONS

The knowledge of how an object is positioned is a precondition for many tasks. In the following we will discuss several applications where pose estimation is a key component.

ROBOTICS

In recent years robots found their way into domestic environments fulfilling many tasks automatically. Such robots mostly fulfill simple cleaning or gardening tasks. However, moving in such environments requires the robot to be aware of its surroundings to be able to fulfill such tasks. The robot is required to create a map of the environment in order to be able to navigate autonomously. This can be achieved by exploring the environment while simultaneous tracking the robots position and creating a map. This concept, known as SLAM (simultaneous mapping and localization), was first introduced by Leonard and Wyhte [67]. In cases when the tracking fails, the robot requires to re-localize itself within the environment enable the contination of the process. Camera re-localization methods [108, 10, 9] can be employed to estimate the current position of the robot and in case of failures reinitialize the tracking process. Those methods require an offline training step which limits their applicability for online processes. In contrast, the method proposed by Cavallari et al. [15] can be trained during the tracking process.

Pose estimation is furthermore applied in the domain of autonomously driving cars. In order to facilitate safe navigation positions of other traffic participants need to be determined constantly to avoid accidents which can cause serious, live-threatening damages.

Apron Monitoring Robotic Grasping & Navigation

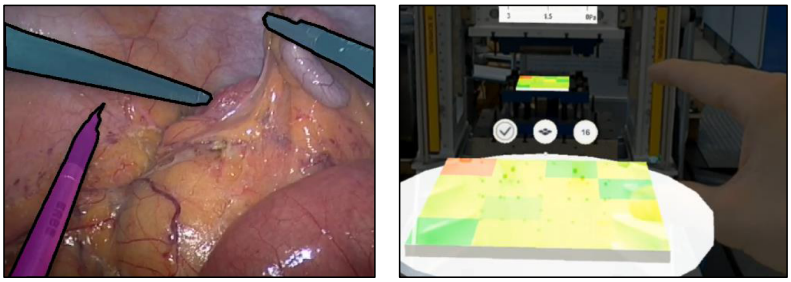
Figure 1.5.: **Applications: Monitoring and Robotics. Left:** A point cloud captured at the apron of the Dresden airport shown in the sensor perspective (top) and the birds eye perspective. The pose of an Airbus 319 aircraft is estimated and shown as the projected object model . **Middle:** Object grasping is performed by a robotic arm. **Right:** A mobile robotic platform autonomously navigating within an indoor environment.

MONITORING

Security and safety is crucial in high dynamic environments like factories, warehouses and transportation hubs where minor mistakes can lead to delays or even accidents that places the health and safety of humans at risk. An apron at the airport is such an unstructured and high dynamic working environment where a large variety of objects interact. However, occurring objects are often known a priori and their number is limited in such facilities. Those environments are furthermore often under surveillance which enables object pose estimation, e.g. aircraft and ground vehicles at an airport apron, employing the observing sensors. Such information can be utilized in a apron monitoring system to prevent high risk situations.

AUGMENTED REALITY

Augmented reality (AR) describes the process of enriching views of the real world with additional information. Most modern mobile phones are equipped with a display and a camera which are the key components to enable augmented reality. Two potential AR use cases could be a tourist and a maintenance guidance system. Camera re-localization could be used to determine the position of tourist within a city center. The AR application would provide additional information, e.g. historical information, about visible buildings based on the estimated camera position. Object pose estimation could provide poses of objects, e.g. parts of a car engine, which would guide a mechanic during maintenance procedures.



Medicine Augmented Reality

Figure 1.6.: **Applications: Medicine and Augmented Reality. Left:** Surgical instruments being detected within an endoscopic camera image. **Right:** Visualization of forces being applied to a workpiece using augmented reality techniques.

MEDICINE

Advances in medical surgery enabled minimal invasive surgery techniques where, in contrast to conventional surgery, only small incisions are required to perform an operation. Applying minimal invasive surgery reduces pain, the risk of infections, and allows for faster recovery of the patients. Surgeons do not have direct view but employ imaging sensors, e.g. endoscopic cameras, to inspect the operation areas. Difficulties for surgeons arise through limited movement flexibility and a restricted field of view. Robotic assistance systems have been developed to reduce those difficulties. Kenngott et al. [59] and Speidel et al. [110] developed methods to guide and support surgeons during the operation. Providing the surgeon with positions of organs or the surgical robot, employing pose estimation technices, and an inner-body navigation system, employing camera re-localization techniques, could furthermore decrease potential risks of the operation process.

1.2. OVERVIEW

Our work is inspired by the previously introduced hypothesize and test strategy which we employ to solve the task of pose estimation. Given an image of the object we need to establish correspondences between positions in the image and positions on the object. This is accomplished using distinctive textural and geometrical features of the object.

We use the concept of object coordinate regression, proposed by Brachmann et al. [8], to solve the correspondence problem. They use machine learning techniques to generate object specific descriptors capturing individual object properties. We contributed to this work, which is a foundation for all the systems presented throughout this thesis and we will introduce the concept in detail in Section 2.2.2.

Each of the proposed methods use depth images as input data. The established object-to-image correspondences therefore relates 3D positions on the object to 3D positions in the image. Pose calculation techniques have been first proposed by Wahba [121] and Schönemann [102]. The Kabsch algorithm [50] is a popular method enabling correspondence-based pose calculation. Given a minimal set of three correspondences the Kabsch algorithm determines the 3D rotation R and the 3D translation t yielding the 6D object pose H.

The algorithm by Kabsch minimizes the root mean squared deviation when calculation the pose from corresspondences. This metric is heavily influenced by outliers leading to an incorrect pose estimate if one or more correspondences are erroneous. The hypothesize and test strategy counteract this by generating not a single pose hypothesis, but a pool of pose hypothesis using the entire set of correspondences. To determine whether the correspondence triplet contained an outlier, the pose hypothesis is tested and a score is calculated which reflects how well the it aligns with the observed data. This allows the ranking of hypothesized poses enabling the rejection of triplets contaminated by outliers correspondences.

In this work, we aim in exploring the benefits of incorporating geometric knowledge we have about the given task. In particular, we focus on the hypothesis generation step within the pose estimation pipeline, where we incorporate this knowledge to increase the efficiency and robustness of the pose estimation process. In Chapter 2, we address the task of 6D pose estimation of aircraft within a highly dynamic apron monitoring scenario. The hypothesis generation is implemented as a random sampling of correspondence triplets. We employ a guided sampling which improves the efficiency of the hypothesis generation. This is done using the information gathered from the depth value of the first correspondence sample to determine the maximal area occupied by the object in the image.

The aircraft are clearly visible within the scene which enables our method to solve the task efficiently. However, with decreasing numbers of inlier correspondences more sampling iterations are required to generate an outlier-free correspondence triplet. In fact, the amount of sampling iterations increases exponentially rendering random sampling based hypothesis generation inefficient. Occlusions are one cause for decreasing inlier rates. The number of potential correspondences is decreased since the visible surface gets smaller and artificial features, e.g. a textural or geometric change

on the occlusion boundary, are added causing erroneous correspondence predictions.

Pose estimation of articulated object, which are assemblies of rigid parts, is challenging since individual parts can cause large occlusions onto other parts of the object. We refer to this effect as self occlusion. Self occlusion can lead to parts being almost entirely occluded, a closed laptop computer is such an example where the display occludes the laptop body almost completely. However, we have the knowledge of how the two parts of the laptop computer are related to each other in their possible configurations. We incorporate this knowledge within the hypothesis generation by representing the articulated object as a kinematic chain. This enables accurate pose estimation employing random sampling even when object parts are largely occluded. This approach is presented in Chapter 3.

When considering self occlusion, we are able to make assumptions about the relationship between the occluder and the occludee. This is not possible when the object of interest is occluded by an unknown object. The system presented in Chapter 4 addresses the challenge of pose estimation under severe occlusion. We phrase the hypothesis generation as an energy minimization problem which is solved using the graphical model framework. Here, we aim to find inlier correspondences by introducing a geometric check that assesses the quality of the predicted object surface positions visible at a pixel within the image. This check is used in the energy minimization process which outputs image segments, labeling consistent pixels as inliers.

1.2.1. CONTRIBUTIONS

The primary contribution of this thesis is threefold:

- We introduce three different systems for the task of object pose estimation and show that incorporating geometric task knowledge is beneficial.
- We show that our systems are not restricted to a specific object type, we address rigid and articulated objects, a specific scenario, we approach indoor and outdoor pose estimation, a specific sensor type, we utilize Lidar-based and structured light-based sensors, and to the input data type, we use both RGB and depth data.
- We demonstrate the robustness of our systems towards self occlusion and occlusion by unknown objects.

The systems proposed in this thesis defined state-of-the-art at the time of their publication. We created two new datasets (see Appendix A.2.1 and A.2.2), which we made publicly available.

1.3. RELATED WORK

Interest in the the task of pose estimation grew beginning in the early 1990s. Early methods addressing the task predominately used depth sensors capturing the geometry of the object. Hand-engineered descriptors defined on the object shape and the

curvature of an object, e.g. Spin images [49], were mostly used to solve the image-to-object correspondence problem. Those methods were applied on the simplified scenario where the object of interest is staning clear in front of the sensor. The dataset introduced by Mian et al. [75] featured multiple objects occluding each other. While this added a new degree of difficulty to the task of pose estimation it was still far away from a realistic test scenario.

The emergence of augmented reality systems led to an increased interest in pose estimation based on RGB images. Methods working on color images [72, 109] employed hand-engineered descriptors relying on color features, e.g. SIFT [71] and SURF [3], to estimate image-to-object correspondences. However, research interest in the task of pose estimation was somehow limited.

This changed with the release of the Microsoft Kinect, a sensor mainly designed for the Mircosoft Xbox gaming console. The sensor had two major advantages. It provided aligned RGB and depth images, allowing the development of methods using both sensor modalities without the costly process of sensor fusion. Furthermore, in contrast to other depth sensing devices, the Kinect sensor came at a low price which led to a fast prevalence within the research community.

A big step for the field of pose estimation was the release of the dataset by Hinterstoisser et al. [37]. It was the first large dataset addressing a more realistic and complex scenario, where objects are standing in a cluttered environment being occluded by other objects. At the same time, machine learning methods pushed into the field of computer vision increasing the accuracy and the robustness of pose estimation methods. Recently, the focus of pose estimation moved from laboratory environments towards industry relevant scenarios which reflects in new datasets addressing bin picking [94] and robot assembly processes [39, 21].

While methods addressing the task of pose estimation come in different varieties, they mostly follow general task solving strategies. This applies in particular to the hypothesis generation process. In the following, we will introduce the three prevailing strategies for pose hypothesis generation and discuss related works for each of them.

1.3.1. EXHAUSTIVE SEARCH

Exhaustive search is a general problem solving technique. Approaching the task of pose estimation employing a naive exhaustive search strategy would create all possible pose hypotheses. Assuming that 3D-3D correspondences, associating positions in the image to positions on the object, are available, the creation of a hypothesis requires three of correspondences. Exhaustively generating these triplets and evaluating whether they are in consensus with the entirety of the observed data can quickly become prohibitive.

Template-based approaches counteract this behavior. A template describes the appearance of an object or a part of an object under a certain viewpoint. To describe the object as a whole, a set of templates is created by displaying the object under varying orientations and scales. The hypothesis generation is conducted by exhaustively comparing each template with all possible image locations which results in a score reflecting the alignment of the template at the specific image position. Hypotheses are created at positions where the score exceeds a predefined threshold. Due to

the nature of the template capturing the object appearance under a certain viewing angle and scale such a hypothesis constitutes a 6D object pose. Restricting the number of templates used to describe the object enables fast computation times but also increases inaccuracy of the hypothesized poses. Most template-based methods therefore employ a refinement process to accurately align the hypothesis to the observed data. Efficient comparison strategies employ holistic object templates which enables template-based methods to operate very fast. Such holistic approaches, however, are prone to errors in cluttered environments or when the object is occluded.

Early template-based methods employ object contour information extracted from the image. While Huttenlocher et al. [44] used the Hausdorff distance, Olson and Huttenlocher [85] used the Chamfer distance [7] to measure the similarity between the template and an image region. Both methods were sensitive to occlusion and clutter limiting their applicability. Instead of only relying on contour information Dalal and Triggs [18] used the histogram of oriented gradients (HOG) descriptor. The HOG descriptor utilizes image gradients making it less sensitive clutter and occlusions. This descriptor was furthermore a building block of the deformable parts model (DPM) proposed by Felzenszwalb et al. [27] which approached the task of 2D object class detection. This method was later extended by Pepik et al. [90] to conduct 3D object class detection and coarse viewpoint estimation.

Conducting object class detection requires the templates to be flexible towards the variation within the class. This restricts the class-based templates from capturing specific features of individual objects which limits the performance on an instance level. Hinterstoisser et al. [35] approached this problem by creating instance specific templates using RGB-D sensor data. Their LINEMOD templates utilized both RGB gradients as well as object surface normals calculated based on the depth channel which increased the robustness towards clutter and changing lighting conditions. While using a large template set, approximately 2000 templates per object being created through object pose and scale variation, their method provided object detections in real time. In an extension, Hinterstoisser et al. [37] addressed the task of 6D pose estimation using the LINEMOD templates to generate pose hypotheses. The best scoring hypotheses were refined using an ICP algorithm to improve the pose accuracy. As in [35], the template matching function was highly efficient providing pose estimates in a fraction of a second. The LINEMOD templates were created by rendering a 3D model of the object in different orientations ($0° - 360°$ around the object, $0° - 90°$ tilt rotation, $-45° - +45°$ in-plane rotation) and scales (65cm - 115cm). However, a growing number of objects linearly increases in the number of templates which is further amplified when a greater range of object orientations and scales should be covered.

Rios-Cabrera and Tuytelaars [95] proposed a cascaded detection scheme and discriminately learned LINEMOD templates using a support vector machine (SVM). This increased the accuracy decreased the run time of the method. Kehl et al. [57] efficiently matched LINEMOD templates using a hash function resulting in a sub-linear scaling and higher accuracy of the method. Hodan et al. [41] achieved sub-linear complexity employing a cascaded scheme. A salience check was used to determine the objectiveness of an image region where template matching was only condcuted on regions passing this test. The method employs templates defined on 3D point and normal information of pixel triplets and a hashing functions was used to further increase

the performance.

Template-based methods have proven to be applicable for the task of pose estimation. While several challenges, (e.g. scalability and robustness towards clutter), have been addressed, template-based methods are still sensible to object occlusion. Self occlusion, occurring with articulated objects, could be modeled during the template creation process. Covering all possible articulation states would tremendously increase the number of templates. The system presented in Chapter 3 shows how to incorporate the object structure into the pose estimation process enabling hypothesis generation of articulated objects which scales linearly with the number of parts.

Zach et al. [126] conducted hypothesis generation employing exhaustive search without utilizing templates. Their approach uses an occupancy-based shape descriptor to determine object coordinates for each pixel within a given depth image. The quality of a pair of object coordinates is assessed by comparing the euclidean distance in camera coordinate space and in object coordinate space. The hypothesis generation process creates pixel triplets by exhaustively pairing one pixel with two other pixels within a predefined image region. The triplets are not evaluated by estimating and testing the object pose. This is done using the aforementioned geometric check within an efficient belief propagation process. The final pose is found conducting an ICP-inspired refinement and scoring technique to the hypothesis set. In Chapter 4, we use the same geometric check, but instead of applying it on a restricted image region, we use it on the entirety of the image.

1.3.2. VOTING-BASED APPROACHES

While templates are holistic descriptors capturing object features globally, voting-based methods operate locally. Each and every pixel in an image contributes to the hypothesis generation process by casting a vote in a application specific, quantized voting space. Pixels voting for the same position are accumulated and hypotheses are created for the peaks in this prediction space. Duda and Hart [23] introduced the generalized Hough transform, a line detection method, which was later extended by Ballard [2] to feature the detection of arbitrary shapes. Here the hypothesis generation was performed by images pixels voting for the slope and y-intercept for a 2D line.

Gall et al. [31] used Hough voting to created hypothesis for the task of 2D object detection in RGB images. They trained a random forests to determine the object class and the displacement of the image patch relative to the object center. During test time, all patches are aggregated in the Hough space that is parametrized by the object class, object size and the position of the object center. The peak in this voting space provides an object detection and a coarse pose estimate.

Lowe [72] combined SIFT features and Hough voting to conduct 3D object recognition in cluttered RGB images. Instead of approaching object detection by matching SIFT features to individual images, they used sets of features from multiple images to define a viewpoint model. During test time, SIFT features are calculated and clustered employing a Hough voting determining the object pose. Their method is fast and robust to occlusion and clutter. However, SIFT features require rich textural information and therefore fail to provide reliable results for texture-less objects.

Tejani et al. [118] approached the task of 6D pose estimation employing Hough forests. They used LINEMOD templates [36] as split functions in the forest and stored 6D pose votes at the leaf nodes. The voting is conducted as a three stage process to reduce the dimensionality of the problem. While the first voting stage finds clusters based on the 2D position of the object in the image, in the seccond and third stage clustering is conducted based on the 3D translation and 3D rotation. In an extension, Doumanoglou et al. [20], replaced the LINEMOD templates by features learned using an auto-encoder CNN. Kehl et al. [55] also used an auto-encoder CNN to create descriptors. They replaced the random forest by a codebook which relates descriptors to view points on the object. The entries in the codebook are subsequently clustered in three stages, similar to [118].

Drost et al. [22] introduced the concept of point pair features. Their method is working on depth data and employs a voting strategy which is divided into two stages. A point pair feature operates on pairs of 3D points and their associated surface normals and uses distances in Euclidean and angular space to characterize positions on the object surface. The first stage is used to determine if an observed input point is part of the object and to which position on the object surface it refers. This is done by exhaustively pairing this point with all other points and computing the point pair features. Each feature casts a vote into a 2D space, voting for a surface position on the object. A peak in the voting space describes a candidate for a input-to-object correspondence and employing the surface normal furthermore represents a 6D object pose. This process is repeated for each point in the input data providing a large set of pose candidates. The candidate poses are only rough estimates due to quantization and sampling differences between the input data and the object model. The second stage removes incorrect candidate poses and increases the accuracy of the final result. To this end, pose candidates are clustered based on their similarity in rotation and translation. The final result is found by averaging the poses contained in the cluster with the largest support. Hinterstoisser et al. [38] extended the approach by addressing the sensitivity to sensor noise and clutter and occlusion. Instead of exhaustively pairing all points, they only considered pairs lying in a certain neighborhood which is defined by the size of the object. Furthermore, they introduced a soft voting to account for sensor noise and added a pose refinement step to improve the quality of the final output.

The concept of object coordinate regression is similar to the Hough voting approaches [118, 20, 55]. However, instead of every image patch directly casting votes into the high dimensional 6D pose space, object coordinates are an intermediate representation describing a 3D point location on the object surface. We show that object coordinates facilitate efficient pose hypothesis generation through random sampling as well as through energy minimization.

1.3.3. SAMPLING-BASED APPROACHES

Sampling-based hypothesis generation is a three stage process. First, image-to-object correspondences are established using local evidence, e.g. SIFT [71] descriptor based on RGB features or the FPFH-descriptor (fast point feature histogram) [98] using depth data. Secondly, sets of those correspondences are randomly sampled and a pose hypothesis is created. The task of pose estimation requires a minimal set of three

correspondences if they relate 3D information on the object to 3D information in the image, and four correspondences for the case where only 2D information is available in the image. Thirdly, the hypothesized poses are tested providing a score depicting their alignment with the entirety of the input data. The RANSAC algorithm [28] is a popular example for sampling-based hypothesis generation.

Sparse feature based methods ([33, 74]) have shown their ability to provide accurate results for the task of pose estimation. Those approaches extract points of interest and match them based on a RANSAC sampling scheme. Instead of using ad hoc descriptors, Lepetit et al. [68] learned descriptors characterizing view points on the object. They synthesized large numbers of images, extracted keypoints and used statistical classification tools to create a compact descriptor. Employing random sampling based hypothesis generation increased the robustness of the method towards erroneous correspondences.

Phillips et al. [91] proposed a method for pose estimation and shape recovery of transparent objects. They employed a random forest to detect object contours, since transparent objects do not contain reliable texture information. Those edge responses are clustered and random sampling is employed to find the axis of revolution of the object. While they provide accurate poses for this challenging task, the nature of this method is limited to objects being rotational invariant along one axis.

Papazov and Burschka [86] used features defined on 3D point pairs and surface normals. They used a hashing schema to match pairs of observed points to pairs of points on the object surface. They used an octree data structure, which partitions the three-dimensional space, to organize the input data. This enables the efficient search for neighboring data point which was incorporated into a RANSAC-based hypothesis generation.

The task of camera re-localization has also been addressed by sampling-based methods. Shotton et al. [108] approached the task of indoor camera re-localization. They trained a random forest to predict scene coordinates densely for each pixel within a RGB-D image. Those correspondences were used to generate hypotheses following a preemptive RANSAC [82] approach. This is an iterative process, where a fixed number of hypotheses is sampled in the first iteration. The set of hypotheses is evaluated using a score defined on the number of inlier pixels and only the better half of the hypothesis set advances to the next refinement iteration. This procedure is continued until only one hypothesis remains.

In contrast to [108], the method by Sattler et al. [100] conducts RGB camera re-localization on scenes at city scale. Here, a visual vocabulary is used to match features in the image to 3D points in the scene. The descriptors were quantized to account for the size of the environment and the ambiguities created by this quantization were counteracted using a visibility voting. Finally, RANSAC is used to estimate the camera pose from 2D-3D correspondences.

Sparse features are robust to occlusion and facilitate fast hypothesis generation. However, with a shift of the application scenario towards robotics the popularity of RGB-based features decreased since they require sufficiently textured objects. The computational complexity of random sampling is growing exponentially with decreasing numbers of inliers. Object occlusion is a major cause for the decrease of inliers. In Chapter 4, we introduce a system being robust to occlusion by formalizing the hy-

pothesis generation as a global energy minimization process.

1.4. OUTLINE

The body of this thesis is divided into six chapters, the first being this introduction. Chapter 2 presents a system addressing the task of 6D pose estimation of a single object instance. We employ the concept of object coordinate regression to establish image-to-object correspondences and use a random sampling strategy to generate pose hypotheses. In Chapter 3, we introduce our system addressing pose estimation of articulated objects. We show that using the underlying structure of such an object is beneficial for the hypothesis generation process. In Chapter 4, we take a different approach to hypothesis generation. We introduce a geometric check, assessing the quality of object coordinate predictions. The hypothesis generation is formalized as an energy minimization problem in which the geometric check is globally applied. This enables us to correctly estimate the poses of severely occluded objects. Chapter 5 summarizes the current state of pose estimation, the remaining challenges and potential directions of future work. The thesis is closed with the concluding remarks in Chapter 6.

2. 6D POSE ESTIMATION

Contents

2.1. Introduction 34
2.2. Background 35
2.2.1. Decision Trees 35
2.2.2. Object Coordinate Regression 36
2.2.3. Hypothesis Scoring 38
2.3. Method 39
2.3.1. Training Data Generation 39
2.3.2. Pose Estimation by Random Sampling 40
2.4. Experiments 41
2.5. Summary 42

2.1. INTRODUCTION

In this chapter, we approach the task of pose estimation of rigid object instances under the assumption that the object of interest is present in the observed data. The discussed method is part of a monitoring system at the airport in Dresden. The focus of this research project lies in particular on the automation of surveillance procedures at the apron.

Airport ground operations are considered to be significant risk drivers in the aviation sector. Especially the actions that take place on the apron, which is in fact an unstructured working environment with a large variety of objects, substantially contribute to the operational risk. Additionally, various activities of moving aircraft, vehicles, equipment and personnel on a limited space turn the apron into a complex and dynamic system that lends itself to accidents and incidents creating a measurably high risk environment. Current legacy procedures for apron control rely on the direct view with only little automation.

The unique apron characteristics and the (temporary) limitations in the monitoring capabilities inevitably impact the situational awareness of all apron operators. Our system addresses the task of 6D pose estimation of known object instances using a Lidar-based sensor. Providing the information where objects are positioned on the

apron strengthens the situational awareness of apron controllers which improves the safety level of apron operations.

The task of object instance pose estimation has mainly been investigated in the robotics domain. It is a key component for a robot to know which objects are present in the scene and how they are positioned relative to the robot. Pose estimation employing point cloud data predominately focused on applications within controlled environments, like factories or warehouses. Is it feasible to make assumptions about the object surroundings in those scenarios. This reflected in early data sets where objects appearing in domestic environments where free standing in the scene, only occluded by other known objects [75, 76].

The emergence of autonomous driving cars created a new application scenario for object detection and pose estimation. Roads are a highly dynamic environment, where a large variety of objects interact and failures can lead to serious risks. Autonomous driving cars need to detect and estimate the poses of other traffice participants in order to prevent risky situations. Pedestrian detection [111] and detection of cars and bicycles [4] has been conducted in the past. It is sufficient to operate on the level of object classes, e.g. cars, and coarsely determine the orientation of other objects. The distance needs to be estimated accurately to assure safely operating autonomous cars.

In contrast to the aforementioned methods, the application scenario of apron monitoring, which is a highly unstructured environment, requires accurate pose estimates on the level of object instances.

CONTRIBUTIONS

- We present a method approaching the task of object pose estimation within the highly dynamic apron environment.
- We show that our system is beneficial for the apron monitoring task by providing pose estimates which fulfill the demanded accuracy defined by a standardization organization.

2.2. BACKGROUND

In this section, we will review the foundations of decision forests and introduce their utilization to solve the correspondence problem in our pose estimation pipeline. Furthermore, we will discuss a scoring function that is employed for the pose hypotheses evaluation.

2.2.1. DECISION TREES

A decision tree is a machine learning technique that composes simple functions to solve a complex task. Those functions, usually call test or split functions, are hierarchically organized within a tree structure (see Figure 2.1). A tree is a special case of a graph. Nodes within the tree only have one incoming connection, except for the root

node, and do not contain cycles. Furthermore, the nodes within the tree can be distinguished into inner and leaf nodes. The inner nodes, also called split nodes, contain the test functions. The results of those tests determine the route through the hierarchical structure and here we only consider tests providing a binary output. A leaf node, also called terminal node, defines the end of such a path and either stores a classification or a regression result.

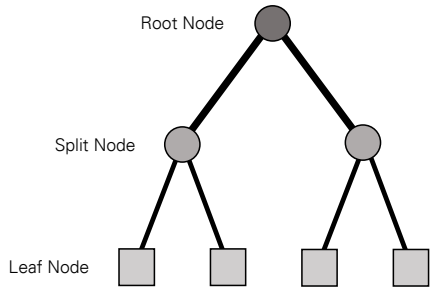


Figure 2.1.: **Decision Tree.** A decision tree is a hierarchically organized structure of nodes and edges. It contains one root node, multiple inner nodes (split nodes), and multiple leaf nodes (terminal nodes).

Two operation modes, a training and a testing mode, of a decision tree can be distinguished. The training of the random forest requires a labeled set of training data. The training process follows a top-down approach starting at the root node where several split functions are evaluated on the entirety of the training data. A metric, e.g. information gain or variance reduction [12], is used to determine the best split function and this process is carried on the child notes where new split functions are evaluated on the subsets of the training data. This learned structure of split functions is used in the test mode where a new data point is pushed through the tree arriving at a leaf node that either provides a class label (classification result) or a real numbered value (regression result). Breiman [12] proposed the combine multiple decisions trees into a decision forest. Randomizing the training procedure by providing a randomly sampled subset of the training data to each individual tree improved generalization capabilities and robustness of the thereby created random forest [11].

2.2.2. OBJECT COORDINATE REGRESSION

Our method is based on the work proposed by Brachmann et al. [8] that uses a random forest to jointly predict the class, determining the object identity, a pixel belongs to and an image-to-object correspondence.

Object coordinates were introduced by the authors to solve the correspondence problem. An object coordinate represents a 3D surface point living within the inherent coordinate system of the object. Figure 2.2 shows object coordinates of an aircraft object where the 3D coordinates are mapped to the RGB cube. Brachmann et al. train a random forest to predict object coordinates given a RGB-D image. This provides

a correspondence between pixels i in the image I and positions on the object $\mathbf{y}_c$. The second output can be understood as a soft segmentation mask. It describes the likelihood for an object c to be present at the pixel location and is therefore called object probability $p_c(i)$. Object probabilities from different trees T are combined into a single value utilizing Bayes' rule.

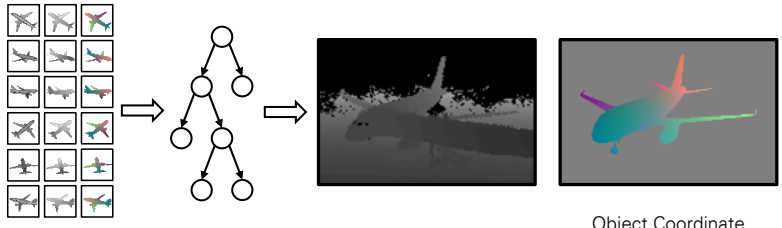


Figure 2.2.: **Object Coordinate Regression.** The random forest is trained on images showing the object under varying poses (left). Those images are created synthetically employing a ray-tracing-based approach to simulate the sensor. Given a depth image, the random forest provides object coordinate predictions for each pixel (right).

The training procedure is divided into two steps. The structure of the forest is learned using a proxy classification task. Proxy classes are defined by discretizing the object coordinate space into $5 \times 5 \times 5 = 125$ bins, adding one additional bin to account for the background class. This enables the use of the standard information gain objective to determine the best split functions. After finalizing the tree structure pixels from all objects are pushed through the tree arriving at the leaf nodes where object coordinates belonging to the same object are clustered. A mean-shift algorithm is performed on the object coordinate distribution to determine the object coordinate that will be associated with that leaf node. Object probabilities associated with a leaf node are calculated through accumulating the incoming pixels of an object.

Feature tests of the split nodes operate on a local image patch level. Evaluating a feature at a pixel location is performed by selecting two other pixels in the vicinity and calculating differences of the assigned input channel, color or depth. Such features are fast to compute which follows the idea of decision trees being an ensemble of simple functions. Depth invariance of the features is achieved following [107] by scaling the patch size depending on the depth value at the pixel.

The training dataset consists of images showing the object under varying poses without containing background information. Color-based feature tests reaching outside the object are modeled by returning random noise. Depth-based feature tests are treated differently utilizing prior task knowledge that the object is standing on a planar surface. The background is therefore modeled by simulating a plane below the object. The training process is randomized by providing each tree with a different set of randomly sampled image pixel location.

2.2.3. HYPOTHESIS SCORING

As introduced in 1.1 the hypothesis evaluation is a key ingredient of the pose estimation pipeline. It allows the assessment of how well the hypothesis aligns with the observed data and facilitates the ranking of hypotheses. We will discuss the hypothesis scoring function that was proposed by Brachmann et al. [8] which is employed in the methods presented in Chapter 2.3 and in Chapter 3.3.

Brachmann et al. phrase the pose estimation process as an energy minimization problem. The energy calculation is performed by comparing observed data, e.g. a depth image captured by a sensor, with synthetically generated data, e.g. a depth image created by rendering the 3D object model under the hypothesized pose. The comparison is conducted on a pixel level punishing derivations between the two compared images. The energy is composed of three different components

$$\hat{E}_c(H_c) = \lambda^{\text{depth}} E_c^{\text{depth}}(H_c) + \lambda^{\text{coord}} E_c^{\text{coord}}(H_c) + \lambda^{\text{obj}} E_c^{\text{obj}}(H_c). \tag{2.1}$$

The depth component is defined as:

$$E_c^{\text{depth}}(H_c) = \frac{\sum_{i \in M_c(H_c)} f(d_i, d_i^*(H_c))}{|M_c(H_c)|}, \tag{2.2}$$

and compares the observed camera coordinates $f(d_i)$, calculated from the depth sensor image using the intrinsic camera parameters, and the rendered camera coordinates $d_i^*(H_c)$ under the hypothesized pose H_c. To account for inaccuracies of the 3D object model a robust function: $f(d_i, d_i^*(H)) = \min\left(||\mathbf{x}(d_i) - \mathbf{x}(d_i^*(H))||, \tau_d\right)/\tau_d$, with τ_d being a free threshold parameter, is employed. The evaluation is performed only on the set of pixel M_c belonging to the object c.

The same strategy is applied for the object coordinate component. It is defined as

$$E_c^{\text{coord}}(H_c) = \frac{\sum_{i \in M_c(H_c)} \sum_{t=1}^{m} g(\mathbf{y}_{i,c}^T, \mathbf{y}_{i,c}(H_c))}{|M_c(H_c)|}. \tag{2.3}$$

At each pixel location the deviation between a rendered object coordinate and a predicted object coordinate is measured. This process is performed for all object coordinate predictions provided by each tree T. The comparison is again performed employing the robust function $g(\mathbf{y}_c(l_i^j), \mathbf{y}_{i,c}(H_c)) = \min\left(||\mathbf{y}_{i,c}^t) - \mathbf{y}_{i,c}(H_c)||^2, \tau_y\right)/\tau_y$.

Finally the object component determines how well an ideal object segmentation mask aligns with the object probabilities predicted by the forest and it is defined as

$$E_c^{\text{obj}}(H_c) = \frac{\sum_{i \in M_c(H_c)} \sum_{j=1}^{t} -\log p(c)}{|M_c(H_c)|}. \tag{2.4}$$

Normalization of the energy components can cause instabilities when the number of pixels occupied by the object is low. To address this the energy is set to $E_c(H_c) = \infty$ if the number pixels is below 100.

2.3. METHOD

The concepts described in the last section forms the foundation of the work presented in this chapter. We follow the hypothesize and test strategy (see 1.1) to estimate the 6D pose H_c of a rigid and stationary object $c \in C$ given 3D point cloud data.

We are working in the domain of apron monitoring and consider object instances of aircraft. Object coordinates are employed to solve the correspondence problem. We provide details to the generation of the training data, enabling object coordinate regression via random forests, in Section 2.3.1. Furthermore, we discuss the pose estimation pipeline and provide details of the hypothesis generation process in Section 2.3.2.

2.3.1. TRAINING DATA GENERATION

Sensors based on structured light, like the Microsoft Kinect, struggle when used in an outdoor scenario. Sunlight interferes with the light patterns projected by the sensor making it difficult to capture reliable data. The range of such sensors is furthermore limited to distances below 10 meters. Lidar-based sensors measure the distance of a position in the scene by sending out laser beams. They are less affected by sunlight, and other difficult weather conditions, like rain or fog, and provide measurements in ranges up to 500 meters. Such sensors are therefore well suited for the task of apron monitoring. A Neptec Opal-360 Lidar sensor operates at the airport in Dresden. This sensor employs a single laser beam circulating within a $360°$ horizontal and $45°$ vertical field of view producing 200000 point measurements per second. Those measurements are in contrast to color images not organized on a grid structure which leads to a lack neighborhood relationship information between the individual data points. As explained in Section 2.2.2 in our scenario the split functions contained within the random forest rely on pixel neighborhood relationships. Reprojecting the captured 3D points onto a 2D image plane, creating a depth image, enables us to use the random forest for correspondence estimation.

Training the random forest requires training data showing the object in different poses. Training data can be acquired by annotating captured sensor data or through synthetic generation, e.g. employing a rendering pipeline. We generated training synthetically since real data is difficult to obtain, especially in our application scenario where the objects of interest are large. We gathered 3D models of the objects from the 3D Google Warehouse [119] (see Figure 2.3). An approximation of the sensor model was provided by the sensor manufacturer Neptec. We use a ray tracing based approach to simulate the sensor. Training data is generated by placing the object on a ground plane and virtually casting rays from the sensor into the scene (see Figure 2.3). We only consider viewing positions resting on the objects upper hemisphere and we parametrize the space of viewing positions by three angles: azimuth, elevation and in-plane rotation. To obtain a good coverage we employ slice sampling [80] to acheive equally distibuted the viewpoints on the hemisphere. This results in 630 viewing positions.

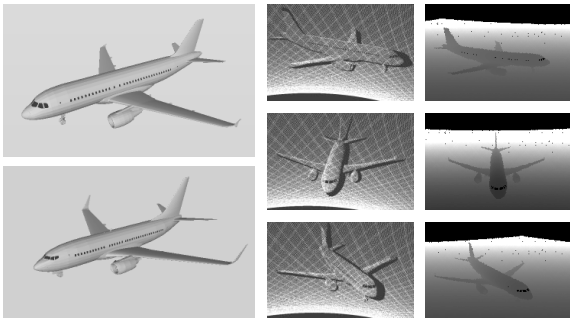


Figure 2.3.: **Training Data Generation.** (a): 3D models of two aircraft. Top: Airbus 319-100. Bottom: Boeing 737-700. (b): Point clouds showing different poses of the Airbus 319-100 that were generated by sensor simulation utilizing a ray casting approach. (c): Depth image representation of the point clouds.

2.3.2. POSE ESTIMATION BY RANDOM SAMPLING

Using the Forest. Once the training in completed we can classify a pixel i by pushing it through the tree. The pixel arrives at a leaf node that stores distributions of object probabilities $p_i(c)$ and object coordinates $\mathbf{y}_c$. Object probabilities from different trees T are combined using Bayes' rule. We use three trees within the random forest leading to three different object coordinate predictions for each pixel.

Hypothesis Sampling. We follow the general idea of RANSAC [28] by creating pose hypotheses by randomly sampling minimal correspondence sets. For the task of 6D pose estimation a minimal set consists of three correspondences between camera coordinates measured by the sensor and object coordinates predicted by the random forest. We utilized the object probability predictions of the random forest to concentrate the sampling on regions within the image that are likely to contain the object.

To start the hypothesis sampling we pick a first pixel i_1 within the image based on a weight proportional to the object probabilities $p_i(c)$ (see Figure 2.4 (c)). We obtain an object coordinate prediction $y(i_1)$ for the pixel by randomly selecting a tree (see Figure 2.4 (d). Together with the world coordinate $x(i_1)$ at the pixel i_1 we obtain the first 3D-3D correspondence $(x(i_1), y(i_1))$ between the object coordinate space and the world coordinate space. We use task knowledge to guide the sampling of the other two correspondences. The first pixel i_1 provides us with a depth measurement that is utilized to determine the object size in the image by projecting the object diameter into the image. This information is used to resrict the sampling of the other two correspondences to lie within this image region which completes the minimal correspondence set. This enables the estimation of H_c using the algorithm by Kabsch [50]. The sampling process is repeated multiple times to account for senor noise and erroneous random forest predictions which creates a pool of hypotheses.

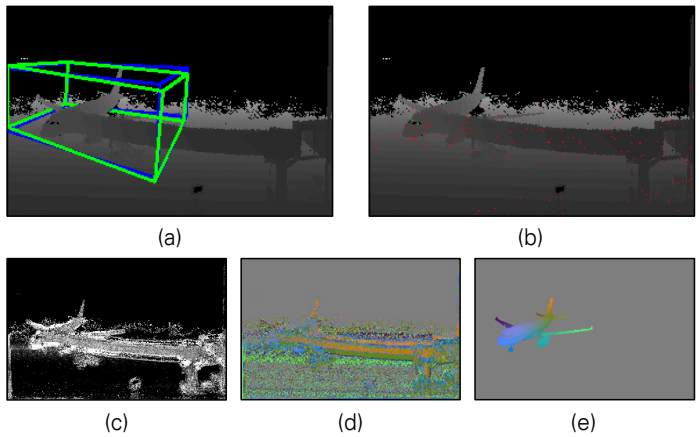


Figure 2.4.: **System Overview.** (a): Depth image representation of a 3D scan showing an Airbus 319-100 in parking position. The ground truth 6D pose is depicted by the blue bounding volume. The estimated pose is shown in green. (b): Hypothesis generation process. Sampled hypotheses are visualized by projecting the first sampled pixel into the image. Color coding: Hypotheses in red color were rejected by a geometric check. Hypotheses in green were refined. (c): Probability map for the query object. (d): Object coordinate prediction for the query object. (e): Ground truth object coordinates.

We select the best hypothesis based on a scoring function similar to what was described in 2.2.3. To evaluate the hypothesis we create a synthetic scan of the object under the hypothsized pose. The score is calculated by a pixel-wise comparison of the depth values from the sensor, the forest predictions and the synthetically created images. We refine the best five hypotheses by recalculating the pose using the inlier set estimated by the scoring function. Inliers are determined by object coordinate component. The pixel is considered to be an inlier, if the distance between the object coordinate in the synthetically generated image and one of the random forest predictions is smaller than 20cm. This evaluation and refinement process is repeated eight times and the hypothesis with the best score is chosen as the final pose.

2.4. EXPERIMENTS

In this section we demonstrate the technical feasibility of our pose estimation method. A proof of concept is conducted using two types of aircraft: an Airbus A319-100 and a Boeing B737-700. We selected both models because of their widespread presence worldwide [1] ensuring a high degree of practical relevance for our test and because of

[1] A319 (A320 family total) delivered: 1454 (6932); B737-700 (737 family total) delivered: 1140 (8929)

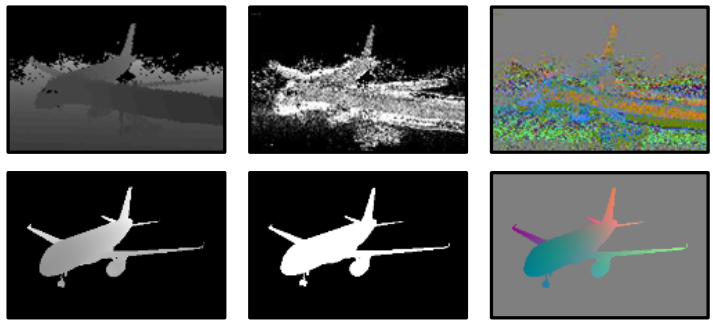


Figure 2.5.: **Hypothesis Scoring.** To determine the quality of a pose hypothesis we use the scoring function introduced in Section 2.2.3. The object model is rendered under the hypothesized pose in three different modalities (depth, object probabilities, and object coordinates). A pixel-wise comparison of the rendered images and the observed images is conducted to calculate the score.

their similar geometrical shape and dimensions to provide a challenging scenario to the algorithm. The data recording was done using the aforementioned Neptec OPAL 360 LiDAR sensor which is installed at the terminal building at the Dresden airport. Both aircraft were recorded in parking position at the same apron stand. The ground truth was annotated manually.

The quality of the predicted poses is assessed by measuring the rotational difference and the translational difference to the ground truth pose. We achieved an angular error of $1.159°$ and a translational error of $0.868m$ for the estimated pose of the Airbus A319-100 aircraft and an angular error of $3.701°$ and a translational error of $0.534m$ for the estimated pose of the Boeing A737-700 aircraft. The results are visualized in Figure 2.6 where the depth image representation of each aircraft model is shown.

Even though we could not find reference values for pose estimation tasks in the field of autonmous driving, we judge these results as accurate from the computer vision perspective. This is also reflected by the plausible bounding volume positions depicted in Figure 2.6. From the Air Traffic Management (ATM) perspective the following can be stated with regard to our stationary aircraft case: The translational error component of the achieved position accuracy is far lower than the position accuracy required by the ICAO A-SMGCS concept (7.5m for stationary/moving aircraft on the movement area [45]).

2.5. SUMMARY

In this chapter, we proposed a system that addresses the task of 6D pose estimation of two aircraft instances within an apron monitoring scenario. We used a random for-

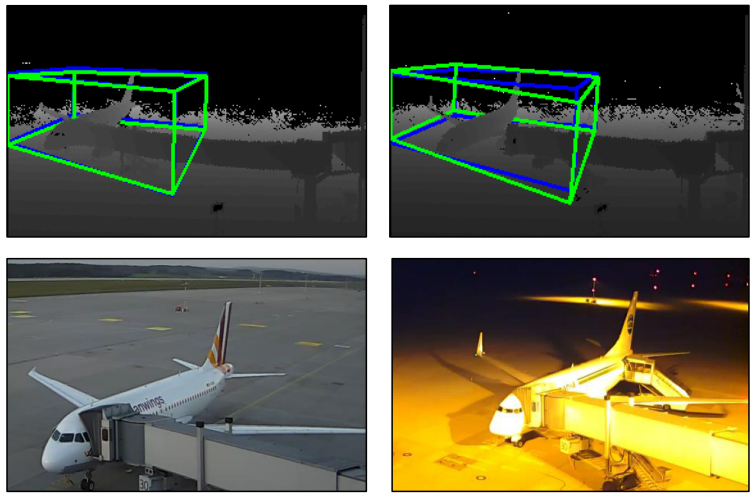


Figure 2.6.: **Results - Sampling-based Pose Estimation.** Estimated poses for the Airbus A319 (left) and B737-700 (right) are shown as green bounding boxes in the depth image in the top row. The two RGB images in the bottom row are not used within our pipeline and are only presented for visualization.

est to solve the image-to-object correspondence problem by predicting object surface points visible at a pixel location. The hypothesis generation was implemented as a RANSAC-based optimization process which has proven to provide accurate results, satisfying standardized position accuracy, in fast run time.

We modelled the two aircraft as rigid objects which is a simplification of the problem since both objects contain movable parts. This did however not impair the accuracy of the estimated poses since those movable parts did not draw occlusions onto other parts. For other objects, e.g. a closet with drawers and doors, parts can cause large occlusions onto each other that will impair the performance of our method. In the next chapter we discuss an extension of the pose estimation pipeline that addresses the challenge of self occlusion when estimating the pose of articulated objects.

3. ARTICULATED POSE ESTIMATION

Contents

3.1. Introduction 44
3.2. Related Work 46
3.3. Method 47
3.3.1. The Articulated Pose Estimation Task 47
3.3.2. Object Coordinate Regression 48
3.3.3. Hypothesis Generation 48
3.3.4. Energy Optimization 50
3.4. Experiments 50
3.4.1. Dataset 50
3.4.2. Setup 51
3.4.3. Results 53
3.5. Summary 55

3.1. INTRODUCTION

Accurate pose estimation of object instances is a key aspect in many applications, including augmented reality or robotics. A task for a robot in a domestic environment could be to fetch an item from an open drawer. The poses of both, the drawer and the item, have to be known in order to fulfill the task. 6D pose estimation of rigid objects has been addressed with great success in recent years. In large part, this has been due to the advent of consumer-level RGB-D cameras, which provide rich and robust input data. However, the practical use of state-of-the-art pose estimation approaches is limited by the assumption that objects are rigid. In cluttered, domestic environments this assumption does often not hold. Examples are doors, many types of furniture, certain electronic devices and toys. A robot might encounter these items in any state of articulation.

This work considers the task of one-shot pose estimation of articulated object instances from an RGB-D image. In particular, we address objects with the topology of a kinematic chain of any length, i.e. objects are composed of a chain of parts interconnected by joints. We restrict joints to either revolute joints with 1 DOF (degrees of

freedom) rotational movement or prismatic joints with 1 DOF translational movement. This topology covers a wide range of common objects (see our dataset for examples). However, our approach can easily be expanded to any topology, and to joints with higher degrees of freedom.

To solve the problem in a straight forward manner one could decompose the object into a set of rigid parts. Then, any state-of-the-art 6D pose estimation algorithm can be applied to each part separately. However, the results might be physically implausible. Parts could be detected in a configuration that is not supported by the connecting joint, or even far apart in the image. It is clear that the articulation constraints provide valuable information for any pose estimation approach. This becomes apparent in the case of self occlusion, which often occurs for articulated objects. If a drawer is closed, then only its front panel is visible. Nevertheless, the associated cupboard poses clear constraints on the 6D pose of the drawer. Similarly, distinctive, salient parts can help to detect ambiguous, unobtrusive parts.

Two strains of research have been prevalent in recent years for the task of pose estimation of rigid objects from RGB-D images. The first strain captures object appearance dependent on viewing direction and scale by a set of templates. Hinterstoisser et al. have been particularly successful with LINEMOD [37]. To support articulation, templates can be extracted for each articulation state. In this case, the number of templates multiplies by the number of discrete articulation steps. The multiplying factor applies for each object joint making this approach intractable with a few parts already.

The second strain of research is based on machine learning. Brachmann et al. [8] achieve state-of-the-art results by learning local object appearance patch-wise. Then, during test time, an arbitrary image patch can be classified as belonging to the object, and mapped to a 3D point on the object surface, called an object coordinate. Given enough correspondences between coordinates in camera space and object coordinates the object pose can be calculated via the Kabsch algorithm. A RANSAC schema makes the approach robust to classification outliers. The approach was shown to be able to handle textured and texture-less objects in dense clutter. This local approach to pose estimation seems promising since local appearance is largely unaffected by object articulation. However, the Kabsch algorithm cannot account for additional degrees of freedom, and is hence not applicable to articulated objects.

In this work, we combine the local prediction of object coordinates of Brachmann et al. with a new RANSAC-based pose optimization schema. Thus, we are capable of estimating the 6D pose of any kinematic chain object together with its articulation parameters. We show how to create a full, articulated pose hypothesis for a chain with K parts from K correspondences between camera space and object space (a minimum of 3 correspondences is required). This gives us a very good initialization for a final refinement using a mixed discriminative-generative scoring function.

CONTRIBUTIONS

- We present a new approach for pose estimation of articulated objects from a single RGB-D image. We support any articulated object with a kinematic chain topology and 1 DOF joints. The approach is able to locate the object without prior segmentation and can handle both textured as well as texture-less objects. To

the best of our knowledge there is no competing technique for object instances. We considerably outperform an extension of a state-of-the-art object pose estimation approach.

- We propose a new RANSAC-based optimization schema, where K correspondences generate a pose hypothesis for a K-part chain. A minimum of 3 correspondences is always necessary.
- We contribute a new dataset consisting of over 7000 frames annotated with articulated poses of different objects, such as cupboards or a laptop. The objects show different grades of articulation ranging from 1 joint to 3 joints. The dataset is also suitable for tracking approaches (although we do not consider tracking in this work).

3.2. RELATED WORK

In the following, we review three related research areas.

ARTICULATED INSTANCES

Pellegrini et al. [88] extended the iterative closest point algorithm to articulated objects. This approach, however, requires at least on a rough pose estimate which limits the applicability of their method for the task of pose estimation. Pauwels et al. [87] presented a tracking framework which incorporates a detector to re-initialize parts in case of tracking failure. Re-initialization without prior information of the object pose, e.g. one shot pose estimation, was not shown. Furthermore, the approach relies on color-based key point detectors which tends to fail on texture-less objects. Sturm et al. [112] and Katz et al. [54] approached the automatic generation of articulated models given an image sequence of an unknown object. These approaches rely on active manipulation of the unknown object and observing its behavior, whereas our work considers one-shot pose estimation of an already known objects.

ARTICULATED CLASSES

In recent years, two specific articulated classes have gained considerable attention in the literature: human pose estimation [117, 107] and hand pose estimation [104, 92]. Some of these approaches are based on a discriminative pose initialization, followed by a generative model fit. Most similar to our work is the approach of Taylor et al. [117] in which a discriminative prediction of 3D-3D correspondences is combined with a non-linear generative energy minimization. Their approach assumes that an accurate segmentation can be obtained through background substraction which is not possible in our scenario. All class-based approaches are specifically designed for the class at hand, e.g. using a fixed skeleton with class-dependent variability (e.g. joint lengths) and infusing pose priors. We consider specific instances with any kinematic chain topology. Pose priors are not necessary.

INVERSE KINEMATICS

In robotics, the problem of inverse kinematics also considers the determination of articulation parameters of a kinematic chain (usually a robotic arm). However, the problem statement is completely different. Inverse kinematics [123, 1] aims at solving a largely underconstrained system for joint parameters given only the end effector position. In contrast, we estimate the pose of a kinematic chain, given observations of all parts.

3.3. METHOD

We will first give a formal introduction of the pose estimation task for kinematic chains (Section 3.3.1). Then we will continue to describe our method for pose estimation, step by step. Our work is inspired by Brachmann et al. [8]. While our general framework is similar, we introduce several novelties in order to deal with articulated objects. The framework consists of the following steps. We use a random forest to jointly make pixel wise predictions: *object probabilities* and *object coordinates*. We will discuss this in Section 3.3.2. We utilize the forest predictions to sample pose hypotheses from 3D-3D correspondences. Here we employ the constraints introduced by the joints of articulated objects to generate pose hypotheses efficiently. We require only K 3D-3D point correspondences for objects consisting of K parts (a minimum of 3 correspondences is required) (Section 3.3.3). Finally, we use our hypotheses as starting points in an energy optimization procedure (Section 3.3.4).

3.3.1. THE ARTICULATED POSE ESTIMATION TASK

In the following, we will describe the task of pose estimation for a kinematic chain. A kinematic chain is an assembly of K rigid parts connected by articulated joints. We denote each part with an index $k \in \{1, \dots, K\}$. We will only consider 1 DOF (prismatic and revolute) joints. A drawer, that can be pulled out of a wardrobe is an example of a prismatic joint. A swinging door is an example of a revolute joint. To estimate the pose of a kinematic chain $\hat{H} = (H_1, \dots, H_K)$ we need to find the 6D pose H_k for each part k. The problem is however constrained by the joints within the kinematic chain. Therefore, we can find the solution by estimating one of the transformations H_k together with all 1D articulations $\theta_1 \dots, \theta_{K-1}$, where θ_k is the articulation parameter between part k and $k+1$. The articulation parameter can be the magnitude of translation of a prismatic joint or the angle of rotation of a revolute joint. We assume the type of each joint and its location within the chain to be known. Additionally, we assume the range of possible articulation parameters for all joints to be known. Given θ_k we can derive the rigid body transformation $A_k(\theta_k)$ between the part k and $k+1$. The transformation $A_k(\theta_k)$ determines the pose of part $k+1$ as follows: $H_{k+1} = H_k A_k(\theta_k)^{-1}$. We can use this to estimate the 6D poses of all parts and thus the entire pose $\hat{H}$ of the chain from a single part pose together with the articulation parameters.

3.3.2. OBJECT COORDINATE REGRESSION

We use the concept of object coordinate regression (see Section 2.2.2) to produce two outputs for each pixel i. Given the input depth image, each tree in the forest predicts object probabilities and object coordinates for each separate object part k of our training set. Object probabilities from all trees are combined for each pixel using Bayes rule. The combined object probabilities for part k and pixel i are denoted by $p_k(i)$. Furthermore, we we obtain multiple object coordinate predictions $\mathsf{y}_k(i) = (\mathsf{x}_k, \mathsf{y}_k, \mathsf{z}_k)^\top$ for each tree, object part k and pixel i. The terms x_k, y_k, and z_k shall denote the coordinates in the local coordinate system of part k. We adhere exactly to the training procedure of introduced in Section 2.2.2 but choose to restrict ourselves to depth difference features for robustness.

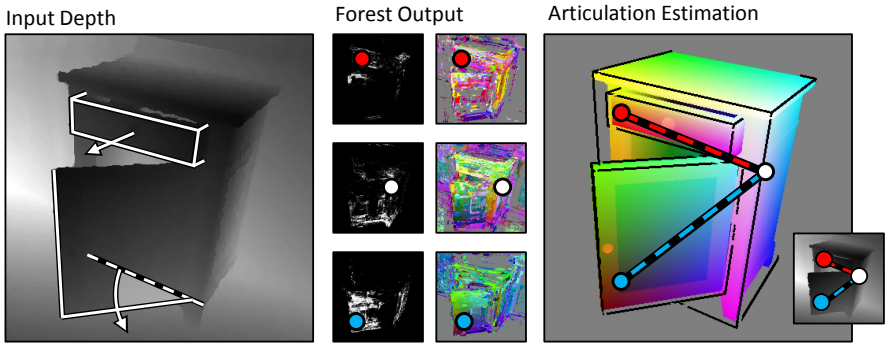


Figure 3.1.: **Articulation Estimation.** Left: Input depth image, here shown for the cabinet. The drawer is connected by a prismatic joint and the door is connected by a revolute joint (white lines are for illustration purposes). Middle: Random forest output. Top to bottom: Drawer, base, door, where the left column shows part probabilities and the right the object coordinate predictions, respectively. Right: Articulation estimation between the parts of the kinematic chain using 3D-3D correspondences between the drawer / base and door / base. Note that the three correspondences (red, white, blue) are sufficient to estimate the full 8D pose.

3.3.3. HYPOTHESIS GENERATION

We now discuss our new RANSAC hypotheses generation schema using the forest predictions assuming that $K = 3$. We will consider kinematic chains with $K = 2$ or $K > 3$ at the end of this section. An illustration of the process can be found in Figure 3.1. We draw a single pixel i_1 from the inner part ($k = 2$) randomly using a weight proportional to the object probabilities $p_k(i)$. We pick an object coordinate prediction $\mathsf{y}_k(i_1)$ from a randomly selected tree t. Together with the camera coordinate $\mathbf{x}(i_1)$ at

the pixel this yields a 3D - 3D correspondence $(\mathbf{x}(i_1), \mathbf{y}_k(i_1))$. Two more correspondences $(\mathbf{x}(i_2), \mathbf{y}_{k+1}(i_2))$ and $(\mathbf{x}(i_3), \mathbf{y}_{k-1}(i_3))$ are sampled in a square window around i_1 from the neighboring kinematic chain parts $k+1$ and $k-1$. We can now use these correspondences to estimate the two articulation parameters θ_{k-1} and θ_k between part k and its neighbors.

ESTIMATING ARTICULATION PARAMETERS

We will now discuss how to estimate the articulation parameter θ_k from the two correspondences $(\mathbf{x}(i_1), \mathbf{y}_k(i_1))$ and $(\mathbf{x}(i_2), \mathbf{y}_{k+1}(i_2))$. Estimation of θ_{k-1} can be done in a similar fashion. The articulation parameter θ_k has to fulfill

$$\|\mathbf{x}(i_1) - \mathbf{x}(i_2)\|^2 = \|\mathbf{y}_k(i_1) - A_k(\theta_k)\mathbf{y}_{k+1}(i_2)\|^2, \tag{3.1}$$

meaning the squared Euclidean distance between the two points $\mathbf{x}(i_1)$ and $\mathbf{x}(i_2)$ in camera space has to be equal to the squared Euclidean distance of the points in object coordinate space of part k. Two solutions can be calculated in closed form. A derivation can be found in the Section A.3. In case of a revolute joint with a rotation around the x-axes the solutions are:

$$\theta_k^1 = \text{asin}\left(\frac{d_\mathbf{x} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_k^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2+b^2}}\right) - \text{atan2}(b,a) \quad \text{and}$$
$$\theta_k^2 = \pi - \text{asin}\left(\frac{d_\mathbf{x} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_k^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2+b^2}}\right) - \text{atan2}(b,a). \tag{3.2}$$

where $d_\mathbf{x} = \|\mathbf{x}(i_1) - \mathbf{x}(i_2)\|^2$ shall abbreviate the squared distance between the two points in camera space. Furthermore $a = 2(\mathsf{y}_k\mathsf{z}_{k+1} - \mathsf{z}_k\mathsf{y}_{k+1})$ and $b = -2(\mathsf{y}_k\mathsf{y}_{k+1} + \mathsf{z}_k\mathsf{z}_{k+1})$. It should be noted that, depending on the sampled point correspondences, θ_k^1 and θ_k^2 might not exist in $\mathbb{R}$ and are thus no valid solutions. Otherwise, we check whether they lie within the allowed range for the particular joint. If both solutions are valid we select one randomly. If no solution is valid, the point correspondence must be incorrect and sampling has to be repeated.

In case of a prismatic joint with a translation along the x-axis we can also solve Equation (3.1) in closed form:

$$\theta_k^1 = -\frac{p}{2} + \sqrt{\left(\frac{p}{2}\right)^2 - q} \quad \text{and} \quad \theta_k^2 = -\frac{p}{2} - \sqrt{\left(\frac{p}{2}\right)^2 - q}, \tag{3.3}$$

where $p = 2(\mathsf{x}_{k+1} - \mathsf{x}_k)$ and $q = (\mathsf{x}_k - \mathsf{x}_{k+1})^2 + (\mathsf{y}_k - \mathsf{y}_{k+1})^2 + (\mathsf{z}_k - \mathsf{z}_{k+1})^2 - d_\mathbf{x}$. Solutions for prismatic joints with translations along other axes can be found analogously. We check again whether θ_k^1 and θ_k^2 are valid solutions in the allowed range of parameters in $\mathbb{R}$ and repeat sampling if necessary. The derivations for 3.2 and 3.3 are provided in Section A.3.

POSE ESTIMATION

Once we estimated θ_k and θ_{k+1} we derive $A_k(\theta_k)$ and $A_{k+1}(\theta_{k+1})$ and map the two sampled points $\mathbf{y}_{k+1}(i_2)$ and $\mathbf{y}_{k-1}(i_3)$ to the local coordinate system of part k. We have now three correspondences between the camera system and the local coordinate system of part k, allowing us to calculate the 6D pose H_k using the Kabsch algorithm. The 6D pose H_k together with the articulation parameters yields the pose $\hat{H}$ of the chain.

In case of a kinematic chain consisting of $n > 3$ parts, we start by randomly selecting an inner part k. We recover the 6D pose using the two neighboring parts as described above. Then, we calculate the missing articulation parameters one by one by sampling one correspondence for each part remaining. In case of a kinematic chain consisting of $n = 2$ parts, we draw a single sample from one part and two samples from the other part.

3.3.4. ENERGY OPTIMIZATION

We rank our pose hypotheses utilizing the energy function introduced in Section 2.2.3:

$$\hat{E}(\hat{H}) = \lambda^{depth} E^{depth}(\hat{H}) + \lambda^{coord} E^{coord}(\hat{H}) + \lambda^{obj} E^{obj}(\hat{H}). \tag{3.4}$$

The kinematic chain is rendered under the pose $\hat{H}$ and the resulting synthetic images are compared to the observed depth values (for E^{depth}) and the predicted object coordinates (for E^{coord}). Furthermore E^{obj} punishes pixels within the ideal segmentation mask if they are unlikely to belong to the object. Weights λ^{depth}, λ^{coord} and λ^{obj} are associated with each energy term. The best hypotheses are utilized as starting points for a local optimization procedure. We used the Nelder-Mead simplex algorithm [81] within a general purpose optimization where we refine the 6D pose H_k of part k together with all 1D articulations $\theta_1 \ldots, \theta_{k-1}$ of the kinematic chain. We consider the pose with the lowest energy as our final estimate.

3.4. EXPERIMENTS

To the best of our knowledge there is no RGB-D dataset which fits our setup, i.e. instances of kinematic chains with 1 DOF joints. Therefore, we recorded and annotated our own dataset.

3.4.1. DATASET

We created a dataset of four different kinds of kinematic chains which differ in the number and type of joints. The objects are a laptop with a hinged lid (one revolute joint), a cabinet with a door and drawer (one revolute and one prismatic joint), a cupboard with one movable drawer (one prismatic joint) and a toy train consisting of four parts (four revolute joints).

TEST DATA

We recorded two RGB-D sequences per kinematic chain with the Kinect sensor, resulting in eight sequences with a total of 7047 frames. The articulation parameters are fixed within one sequence but vary between sequences. The camera moved freely around the object, with object parts sometimes being partly outside the image. In some sequences parts were occluded.

Depth maps produced by the Kinect sensor are prone to missing measurements at object edges and on certain materials. This is problematic for the laptop, since there are no measurements for the display which is a large portion of the lid. To circumvent this, we use an off-the-shelf hole filling algorithm by Liu et al. [70] to pre-process all test images.

We modeled all four kinematic chains with a 3D modeling tool and divided each object into individual parts according to the articulation. Ground truth annotation for the parts were created manually, including all articulations, for all test sequences. We manually registered the models of the kinematic chains onto the first frame of each sequence. Based on this initial pose an ICP algorithm was used to annotate the consecutive frames, always keeping the configuration of joints fixed. We manually re-initialized the object pose if the ICP-algorithm failed.

TRAINING DATA

Similar to the setup in [37], we render our 3D models to create training sets with a good coverage of all possible viewing angles. Hinterstoisser et al. [37] used a regular icosahedron-based sampling of the upper object hemisphere. Different levels of in-plane rotation were added to each view. Since our training images always contain all parts of the kinematic chain, more degrees of freedom have to be taken into account, and each view has to be rendered with multiple states of articulation. Therefore, we follow a different approach in sampling azimuth, elevation, in-plane rotation, and articulation to create the images. Since naive uniform sampling could result in an unbalanced coverage of views we chose to deploy a stratified sampling approach. For all kinematic chains we subdivide azimuth in 14, elevation in 7 and the in-plane rotation in 6 subgroups. The articulation subgroups where chosen as follows: Laptop: 4, Cabinet: 3 (door), 2 (drawer), Cupboard: 4, Toy train: 2 for each joint. This results in $14 \times 7 \times 6 \times 4 = 2352$ training images for the laptop object.

3.4.2. SETUP

In this section, we describe our experimental setup. We introduce our baseline and state training and testing parameters.

BASELINE

We compare our method to the 6D pose estimation pipeline of Brachmann et al. [8]. We treat each object part as an independent rigid object and estimate its 6D pose. This drops any articulation or even connection constrains.

TRAINING PARAMETERS

We use the same parameters as Brachmann et al. [8] for the random forest. However, we disabled RGB features because we expect our rendered training data to be not realistic in this regard. On the other hand, to counteract a loss in expressiveness and to account for varying object part sizes, we changed one maximum offset of depth difference features to 100 pixel meters while keeping the other at 20 pixel meters. For robustness, we apply Gaussian noise with small standard deviation to feature responses. In tree leafs we store all modes with a minimum size of 50% with respect to the largest mode in that leaf. Mode size means the number of samples that converged to the mode during the mean-shift process. We train one random forest for all four kinematic chains jointly (11 individual object parts). As negative class we use the background dataset published by Brachmann et al. [8]. As mentioned above, training images contain all parts of the associated kinematic chain. Additionally, we render a supporting plane beneath the object. Features may access depth appearance of the other parts and the plane. Therefore, the forest is able to learn contextual information. If a feature accesses a pixel which belongs neither to plane nor to a kinematic chain part, random noise is returned. We use the same random forest for our method and the baseline.

TEST PARAMETERS

For the baseline we use the fast settings for energy minimization as proposed by [8]: They sample 42 hypotheses and refine the 3 best with a maximum of 20 iterations. We adopt their setup and treat each part of a kinematic chain separately. Our method, in contrast, creates hypothesis by drawing samples from the kinematic chain in its entirety. Therefore, in our method, we multiply the number of hypotheses with the number of object parts (e.g. $2 \times 42 = 84$ for the laptop). Similarly, we multiply the number of best hypotheses refined with the number of parts (e.g. $2 \times 3 = 6$ for the laptop). We stop refinement after 150 iterations.

METRIC

The poses of all parts of the kinematic chain have to be estimated accurately in order to be accepted as a correct pose. We deploy the following pose tolerance [8, 37, 66] on each of the individual object parts k : $\frac{1}{|\mathcal{M}_k|} \sum_{\mathbf{x} \in \mathcal{M}_k} ||H_k\mathbf{x} - \tilde{H}_k\mathbf{x}|| < \tau, k \in \mathcal{K}$, where $\mathbf{x}$ is a vertex from the set of all vertices of the object model[1] $\mathcal{M}_k$, $\tilde{H}_k$ denotes the estimated 6D transformation and H_k denotes the ground truth transformation. Threshold τ is set to 10% of the object part diameter. We also show numbers for the performance of individual object parts. The results are shown in Table 3.4.2 and discussed below.

[1] The vertices of our models are virtually uniform distributed since we created them manually using a 3D modelling tool.

Object	Sequence		Method	
			Brachmann et al. [8]	Our Method
Laptop	1	all	8.9%	64.8%
		parts	29.8% 25.1%	65.5% 66.9%
	2	all	1%	65.7%
		parts	1.1% 63.9%	66.3% 66.6%
Cabinet	3	all	0.5%	95.8%
		parts	86% 46.7% 2.6%	98.2% 97.2% 96.1%
	4	all	49.8%	98.3%
		parts	76.8% 85% 74%	98.3% 98.7% 98.7%
Cupboard	5	all	90%	95.8%
		parts	91.5% 94.3%	95.9% 95.8%
	6	all	71.1%	99.2%
		parts	76.1% 81.4%	99.9% 99.2%
Toy train	7	all	7.8%	98.1%
		parts	90.1% 17.8% 81.1% 52.5%	99.2% 99.9% 99.9% 99.1%
	8	all	5.7%	94.3%
		parts	74.8% 20.3% 78.2% 51.2%	100% 100% 97% 94.3%

Table 3.1.: Comparison of Brachmann et al. [8] and our approach on the four kinematic chains. Accuracy is given for the kinematic chain (all) as well as for the individual parts (parts).

3.4.3. RESULTS

The baseline can detect individual parts fairly well in cases where the level of occlusion caused by other parts of the kinematic chain is moderate to low. An example is the performance for both cupboard sequences (Sequences 5 & 6) as well as the individual performance of the first (locomotive) and the third part of the toy train (Sequences 7 & 8). However, the method is not able to handle strong self occlusion. This can be seen in the poor performance of the last part of the toy train (Sequences 7 & 8) and in the complete failure to estimate the pose of the cabinet drawer when it is only slightly pulled out (Sequence 3), see Figure 3.3 (first row, second column).

Providing contextual information between object parts during the training of the random forest does not seems to be sufficient to resolve the issues caused by self occlusions. Flat objects do not stand out of the supporting plane, which results in noisy predictions of the random forest. This explains the rather poor performance of the second part of the toy train which is almost completely visible within the entire test sequences (Sequences 7 & 8).

Our method shows superior results (89% averaged over all sequences and objects) in comparison to the baseline (29%). Employing articulation constraints within the kinematic chain results in better performance on the individual parts as well as for the kinematic chains in their entirety, see Table 3.4.2. Our approach of pose sampling for kinematic chains does not only need less correspondences, it is also robust in the presence of heavy self occlusion. Even in cases where one part is occluded more than 75%, e.g. the laptop keyboard in Sequence 2, our method is still able to correctly

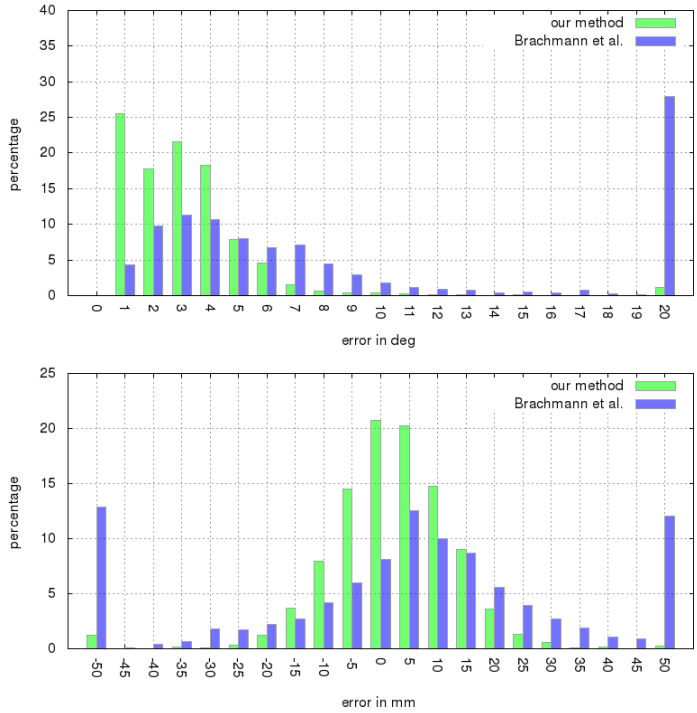


Figure 3.2.: **Comparison of Articulation Estimation.** Histogram of rotational and translational error of our approach compared to [8] for the cabinet (sequence 4)

estimate the pose of the occluded part, see Figure 3.3 (second row, first column). Our approach enables parts with high quality forest predictions to boost neighboring parts with noisy forest predictions (e.g. the second part of the toy train in Sequences 7 & 8).

We furthermore compare our approach to the method of [8] with regard to the error of the articulation parameter. Figure 3.2 shows results for the cabinet in sequence 4. Poses estimated with our method result in a low error for both the prismatic (translational) as well as the revolute (rotational) joint. As a result the distribution for our approach is peaked closely around the true articulation parameter. This is not the case for the approach of [8]. The peak for the rotational error lies at 3° and the peak for the translation lies at +5mm.

3.5. SUMMARY

In this chapter, we presented a system that approaches pose estimation of articulated objects. The inherent structure of the object was represented by a kinematic chain model, which enables our method to efficiently create pose hypotheses and provide accurate results even if object parts are heavily occluded. In case of articulated objects we are able to make assumptions about the object causing the occlusion. This is however not the possible when an unknown object is causing the occlusion. In the next chapter, we will address the challenges created by occlusion through unknown objects.

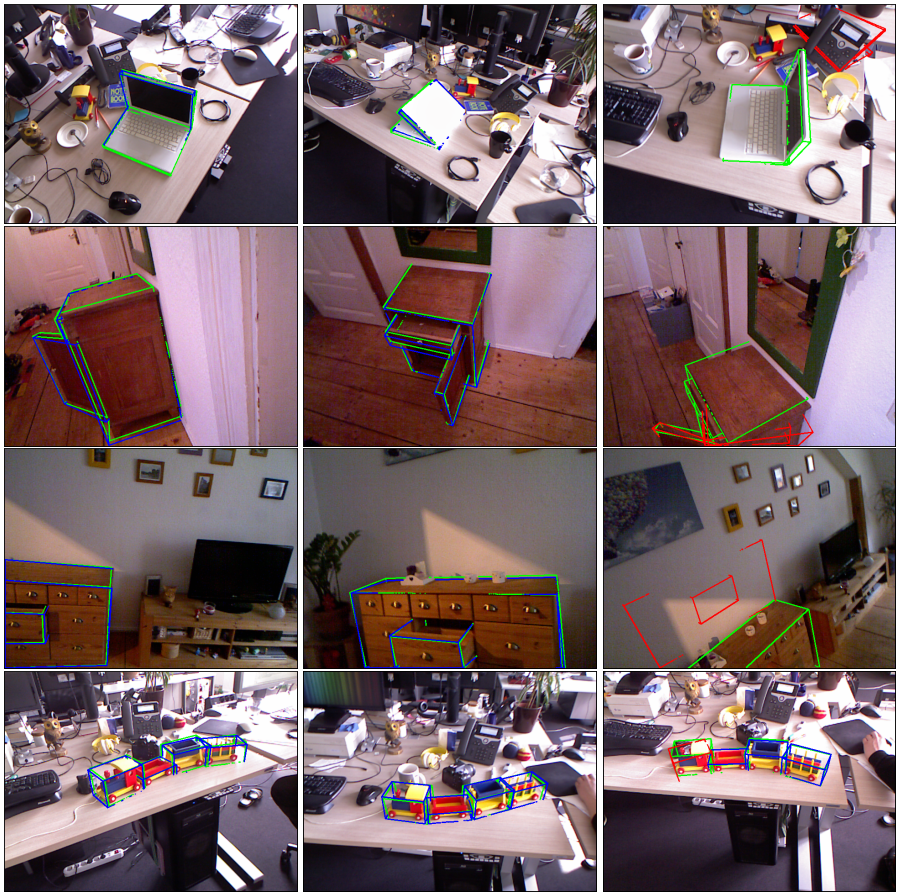


Figure 3.3.: **Results - Articulated Pose Estimation.** These images show results on our dataset. The estimated poses are depicted as the blue bounding volume, the ground truth is shown as the green bounding volume of the object parts. The last row contains failure cases where the bounding boxes of the estimated poses are shown in red.

4. GLOBAL HYPOTHESIS GENERATION

Contents

4.1. Introduction . . . 58
4.2. Related Work . . . 61
4.3. Background - Graphical Models . . . 62
4.4. Method . . . 63
4.4.1. Global Reasoning . . . 63
4.4.2. Method - Graphical Model . . . 64
4.4.3. Energy Minimization . . . 65
4.4.4. Pose Estimation as Energy Minimization . . . 65
4.4.5. Stage One: Problem Size Reduction . . . 67
4.4.6. Stage Two: Generation of Solution Candidates . . . 67
4.4.7. On Optimality of Subproblem Solutions for Binary Energy Minimization . . . 70
4.4.8. Obtaining Candidates for Partial Optimal Labeling . . . 71
4.4.9. Refinement and Hypothesis Scoring . . . 72
4.5. Experiments . . . 72
4.5.1. Dataset . . . 73
4.5.2. Results . . . 73
4.6. Summary . . . 74

4.1. INTRODUCTION

The task of estimating the 6D pose of texture-less objects has gained a lot of attention in recent years. From an application perspective this is probably due to the growing interest in industrial robotics, and in various forms of augmented reality scenarios. From an academic perspective the dataset of Hinterstoisser et al. [37] marked a milestone, since researchers started to benchmark their efforts and progress in research started to be more measurable. In this work we focus on the following task. Given an RGB-D image of a 3D scene, in which a known 3D object is present, i.e. its 3D shape and appearance is known, we would like to identify the 6D pose (3D translation and 3D rotation) of that object.

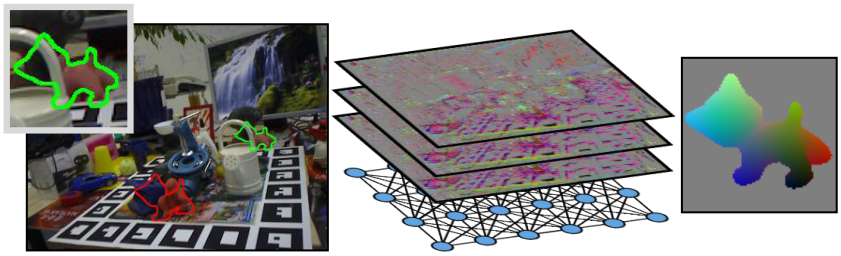


Figure 4.1.: **Motivation.** Given an RGB-D input image (left) we aim at finding the 6D pose of a given object, despite it being strongly occluded (see zoom). Here our result (green) is correct, while Krull et al. [64] outputs an incorrect pose (red). The key concept of this work is to have a *global*, and hence powerful, geometric check, in the beginning of the pose estimation pipeline. This is in stark contrast to *local* geometric checks performed by all other methods. In a first step, a random forest predicts for each pixel a set of three possible object coordinates, i.e. dense continuous part labeling of the object (middle). Given this, a fully-connected pairwise Conditional Random Field (CRF) infers globally those pixels which are consistent with the 6D object pose. We refer to those pixels as *pose-consistent*. The final pose is derived from these pose-consistent pixels via an ICP-variant.

Let us consider an exhaustive-search approach to this problem. We generate all possible 6D pose hypotheses, and for each hypothesis we run a robust ICP algorithm [6] to estimate a robust geometric fit of the 3D model to the underlying data. The final ICP score can then be used as the objective function to select the final pose. This approach has two great advantages: (i) It considers all hypotheses; (ii) It uses a geometric error to prune all incorrect hypotheses. Obviously, this approach is infeasible from a computational perspective, hence most approaches generate first a pool of hypotheses and use a geometrically motivated scoring function to select the right pose, which can be refined with robust ICP if necessary. Table 4.1 lists five recent works with different strategies for "hypotheses generation" and "geometric selection". The first work by Drost et al. [22], and recently extended by Hinterstoisser et al. [38], has no geometric selection process, and generates a very large number of hypotheses. The pool of hypotheses is put into a Hough-space and the peak of the distribution is found as the final pose. Despite its simplicity, the method achieves very good results, especially on the "Occluded Object Dataset"[1], i.e. where objects are subject to strong occlusions. We conjecture that the main reason for its success is that it generates hypotheses from all local neighborhoods in the image. Especially for objects that are

[1] http://cvlab-dresden.de/iccv2015-occlusion-challenge/

Method	Intermediate Representation	Hypotheses Generation	Average Number of Hypotheses	Hypotheses Selection	Hypotheses Refinement	Run Time
Drost et al. [22] Hinterstoisser et al. [38]	Dense Point Pair Features	All local pairs (large neighbourhood)	$\sim$ 20.000	Sub-optimal search	ICP	0.4s
Zach et al. [126]	multiple object coordinates	All local triplets with geometric check	2.000	Optimal w.r.t. PDA	PDA	0.5s
Brachmann et al. [8]	multiple object coordinates	Sampling triplets with geometric check	210	Optimal w.r.t. Energy	ICP variant	2s
Krull et al. [64]	multiple object coordinates	Sampling triplets with geometric check	210	Optimal w.r.t. CNN	ICP variant	10s
Our	multiple object coordinates	Fully-connected CRF with geometric check	0-10	Optimal w.r.t. ICP variant	ICP variant	1-3s

Table 4.1.: A broad categorization of six different 6D object pose estimation methods with respect to four different computational steps: (a) Intermediate representation, (b) Hypotheses generation, (c) Hypotheses selection, (d) Hypotheses refinement, (e) Runtime. The key difference between the methods is marked in red: the number of generated hypotheses. We clearly generate least amount of hypotheses. For this we run an CRF-based hypotheses generation method which is more time-consuming and complex than in other approaches. Please note that our overall runtime is competitive. On the other hand, since we have fewer hypotheses, we can afford a more expensive ICP-like procedure to optimally select the best hypothesis. We show that we achieve results which are superior to all other methods on the challenging "Occluded Object Dataset". (Note PDA stands for "projective data association".)

subject to strong occlusions, it is important to predict poses from as local information as possible. The other three approaches [8, 64, 126] use triplets, and are all similar in spirit. In a first step they compute for every pixel one, or more, so-called object coordinates, a 3D continuous part-label on the given object (see Figure 4.1 right). Then they collect locally triplets of points, in [126] these are all local triplets and in [8, 64] they are randomly sampled with RANSAC. For each triplet of object coordinates they first perform a geometry consistency check (see [8, 64, 126] for details[2]), and if successful, they compute the 6D object pose, using the Kabsch algorithm. Due to the geometric check it is notable that the amount of generated hypotheses is substantially less for these three approaches [8, 64, 126] than for the previously discussed [22, 38]. Due to this reason, the methods [8, 64, 126] can run more elaborate hypotheses selection procedures to find the optimal hypothesis. In [126] this is done via a so-called robust "projective data association" procedure, in [8] via a hand-crafted, robust energy, and in [64] via a CNN that scores every hypothesis. Our work is along the same direction as [8, 64, 126], but goes one step forward. We present a novel, and more powerful, geometric check, which results in even fewer hypotheses (between 0-10). For this reason we can also afford to run a complex ICP-like scoring function for selecting the best hypothesis. Since we achieve results that are better than state-of-the-art on the challenging occlusion dataset, our pool of hypotheses has at least the same quality as the larger hypotheses pool of all other methods. Our geometric check works roughly

[2]For instance, the geometric check of [8, 64] determines whether there exists a rigid body transformation of the triplets of 3D points, given by the depth image, for the triplet of 3D points from the object coordinates.

as follows. For each pair of object coordinates a geometry-consistency measure is computed. We combine a large number of pairs into a fully-connected Conditional Random Field (CRF) model. Hence, in contrast to existing work we perform a *global* geometry check and not a *local* one. It is important to note that despite having a complex CRF, we are able to have a runtime which is competitive with other methods, even considerably faster than [64]. As a side note, we also achieve these state-of-the-art results with little amount of learning, in contrast to e.g. [64].

CONTRIBUTIONS

- We are the first to propose a novel, *global* geometry check for the task of 6D object pose estimation. For this we utilize a fully-connected Conditional Random Field (CRF) model, which we solve efficiently, although its pairwise costs are non-Gaussian and hence efficient approximation techniques like [63] cannot be utilized.
- We give a new theoretical result which is used to compute our solutions. We show that for binary energy minimization problems, a (partial) optimal solution on a subgraph of the graphical model can be used to find a (partial) optimal solution on the whole graphical model. Proper construction of such subgraphs allows to drastically reduce the computational complexity of our method.
- Our approach achieves state-of-the-art results on the challenging occlusion dataset, in reasonable run-time (1-3s).

4.2. RELATED WORK

Hypothesis Generation based on sampling 1.3.3, on voting 1.3.2 and on exhaustive search 1.3.1 have already been discussed. Therefore we will only review methods based on optimization.

Optimization techniques have been used for object detection since the early days of computer vision. The prominent work of Fischler and Elschlager [29] introduced the concept of pictorial structures. They model the object as a collection of several parts that are arranged in a deformable configuration. The object detection is approached by detecting the individual parts using their visual properties and solving the deformable configuration by minimizing an energy function. Inspired by this work Felzenszwalb et al. [27] proposed the deformable parts model. Their method popularized the part-based approaches as it enabled object detection for classes with significant variations like bicycles. Furthermore, Pepik et al. [90, 89] extended the 2D deformable parts model and applied the concept to solve the task of 3D object detection and coarse viewpoint estimation. Those methods only provide 2D detections or coarse 6D pose estimates of object classes which does not provide the accuracy needed for applications like augmented reality or object grasping.

Winn et al. [124] and Hoiem et al. [42] approached 2D [124] and 3D [42] pose estimation of object categories. They also use the key concept of discretized object coordinates for object detection and pose estimation. The MRF-inference stage for

finding pose-consistent pixels is closely related to ours. Foreground pixels are accepted when the layout consistency constraint (where layout consistency means that neighboring pixels should belong to the same part) is satisfied. However since the shape of the object is unknown, the pairwise terms are not as strong as in our case. The closest related work to ours is Bergholdt et al. [5]. They use the same strategy of discriminatively modeling the local appearance of object parts and globally inferring the geometric connections between them. To detect and find the pose of articulated objects (faces, human spines, human poses) they extract feature points locally and combine them in a probabilistic, fully-connected, graphical model. However they rely on a exact solution to the problem while a partial optimal solution is sufficient in our case. We therefore employ a different approach to solve the task.

The graph matching problem (see e.g. [17, 113]) is another formalism used to find true correspondences from a large number of hypothetic correspondences using geometric constraints. However, one key aspect of graph matching is that one discrete feature (e.g. discrete object coordinate of a 3D model) can only match to one other discrete feature (e.g. discrete object coordinate candidate in the image (output from a decision tree)). Our problem formulation, in contrast, has continuous object coordinates.

4.3. BACKGROUND - GRAPHICAL MODELS

In this section we will give a short introduction to the theory of graphical models which we employ within the hypothesis generation process.

Graphical models are probabilistic models that encode relationships between multiple variables [83]. Their probabilistic nature enables them to offer answers to problems where a correct solution can not be determined with certainty by providing a probability distribution over all possible solutions. Graphical models are most commonly represented by graphs $G = (V, E)$, which contain a set of nodes V and a set of edges E. The nodes, also called vertices, represent random variables and the edges model probabilistic relationships between those nodes. Markov random fields (MRF) are a particular type of graphical models where the edges are undirected. MRFs can be applied to solve image analysis problems which are often posed as labeling problems in which a pixel i within the image I is represented by a node and the solution is determined by assigning a label $l_u \in \mathbb{L}$ to each of the nodes $u \in V$ within the graph G.

The factorization of a Markov random field is achieved by defining potential functions on subsets of nodes within the graph G. Potential functions on single nodes, also called unary potentials ψ_u, and potential functions on pairs of nodes, also called binary potentials ψ_{uv}, are most commonly used. The joint probability distribution $P(l)$ over the graph G is then defined as follows

$$P(l) = \frac{1}{Z} \prod_{u \in V} \psi_u(l_u) \prod_{uv \in E} \psi_{uv}(l_u, l_v), \tag{4.1}$$

where the normalization constant Z is given by

$$Z = \sum_{l \in \mathbb{L}} \prod_{u \in V} \psi_u(l_u) \prod_{uv \in E} \psi_{uv}(l_u, l_v). \tag{4.2}$$

Defining the energy functions $\theta_u = -\log(\psi_u)$ and $\theta_{uv} = -\log(\psi_{uv})$ enables rewriting $P(l)$ as follows

$$P(l) = P(l) = \frac{1}{Z} \exp(-\sum_{u \in V} \psi_u(l_u) + \sum_{uv \in E} \psi_{uv}(l_u, l_v)), \tag{4.3}$$

with the normalization constant

$$Z = \sum_{l \in \mathbb{L}} \exp(-\sum_{u \in V} \theta_u(l_u) + \sum_{uv \in E} \theta_{uv}(l_u, l_v). \tag{4.4}$$

The solution to finding a label $l \in \mathbb{L}$ with the highest probability can now be formalized as an energy minimization problem

$$\underset{l \in \mathbb{L}}{\operatorname{argmax}} P(l) = \underset{l \in \mathbb{L}}{\operatorname{argmin}} \sum_{u \in V} \theta_u(l_u) + \sum_{uv \in E} \theta_{uv}(l_u, l_v). \tag{4.5}$$

4.4. METHOD

Our algorithm consists of three stages (see Figure 4.2). The correspondence problem is approached in the first stage where we use the object coordinate regression concept, introduced in Section 2.2.2, to densely predict object probabilities and object coordinates. The random forest $\mathcal{T}$ consists of three trees and each tree T provides a object coordinate for each pixel i in the image I. The object probabilities from multiple trees are, as described in Section 2.2.2, combined to one value using Bayes rule.

We phrase the hypothesis generation as an energy minimization process where we use a graphical model to globally reason about hypotheses inliers. This second stage of the approach is described in Section 4.4.1 roughly and in Section 4.4.2 in detail. In the final stage (Section 4.4.9) we refine and rank our pose hypotheses to determine the best estimate.

4.4.1. GLOBAL REASONING

In general, to estimate the pose of a rigid object, a minimal set of three correspondences between 3D points on the object and in the 3D scene is required [50]. The 3D points on the object, i.e. in the object coordinate system, are predicted by the random forest. One possible strategy is to generate such triplets randomly by RANSAC [28], as proposed in [8]. However, this approach has a serious drawback: the number of triples which must be generated by RANSAC in order to have at least a correct triple with the probability of 95%, is very high. Assuming that n out of N pixels contain correct correspondences, the total number of samples is $\frac{\log(1-0.95)}{\log(1-(1-n/N)^3)}$. For $n/N = 0.005$,

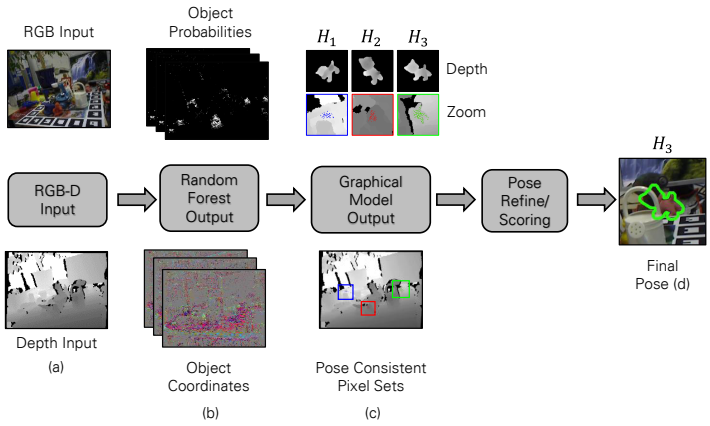


Figure 4.2.: **Global Hypothesis Generation Pipeline:** Given an RGB-D image (a) a random forest provides two predictions: object probabilities and object coordinates (b). In a second stage our novel, fully-connected CRF infers pose-consistent pixel-sets (see zoom) (c). In the last stage, pose hypotheses given by pose-consistent pixels of the CRF are refined and scored by an ICP-variant. The pose with the lowest score is given as output (d).

which corresponds to a state-of-the-art local classifier, this constitutes $\sim$ 24.000.000 RANSAC iterations. Therefore, we address this problem with a different approach. Our goal is to assign to each pixel either one of the possible correspondence candidates, or an "outlier" label. We achieve this by formalizing a graphical model where each pixel is connected to every other pixel with a pairwise term. The pairwise term encodes a geometric check which is defined later. The optimization problem of this graphical model is discussed in Section 4.4.4.

4.4.2. METHOD - GRAPHICAL MODEL

After a brief introduction to graphical models (Section 4.4.3), we define our graphical model used for object pose estimation (Section 4.4.4). This is a fully-connected graph where each node has multiple labels, here 13. The globally optimal solution of this problem gives a pose-consistent (inlier) label to only those pixels that are part of the object, ideally. Since our potential functions are non-Gaussian the optimization problem is very challenging. We solve it approximately, but very efficiently, in a two stage procedure. The first stage conservatively prunes those pixels that are *likely* not inliers. This is done with a sparsely connected graph and TRW-S [61] as inference procedure (Section 4.4.5). The second stage (Section 4.4.6 - 4.4.8) describes an efficient procedure for solving the problem with only the inlier candidates remaining. We prove that by splitting this problem further into subproblems, in a proper way, a (partial) solution to one of these subproblems is guaranteed to be the (partial) optimal solution of

the whole second stage problem. We use the found solutions to the subproblems to generate pose hypotheses.

4.4.3. ENERGY MINIMIZATION

Let $G = (V, E)$ be an undirected graph with a finite *set of nodes* V and *a set of edges* $E \in \binom{V}{2}$. With each node $u \in V$ we associate a finite *set of labels* L_u. Let $\prod$ stand for the Cartesian product. The set $\mathbb{L} = \prod_{u \in V} L_u$ is called *the set of labelings*. Its elements $l \in \mathbb{L}$, called *labelings*, are vectors $l = (l_u \in L_u \colon u \in V)$ with $|V|$ coordinates, where each one specifies a label assigned to the corresponding graph node. For each node *a unary cost function* $\theta_u \colon L_u \to \mathbb{R}$ is defined. Its value $\theta_u(l_u)$, $l_u \in L_u$ specifies the cost to be paid for assigning label l_u to node u. For each two neighboring nodes $\{u, v\} \in E$ *a pairwise cost function* $\theta_{uv} \colon L_u \times L_v \to \mathbb{R}$ is defined. Its value $\theta_{uv}(l_u, l_v)$ specifies compatibility of labels l_u and l_v in the nodes u and v, respectively. The triple $(G, \mathbb{L}, \theta)$ defines *a graphical model*.

The *energy* $E_V(l)$ of a labeling $l \in \mathbb{L}$ is a total sum of the corresponding unary and pairwise costs

$$E_V(l) := \sum_{u \in V} \theta_u(l_u) + \beta \sum_{uv \in E} \theta_{uv}(l_u, l_v)\,. \tag{4.6}$$

Finding a labeling with the lowest energy value constitutes *an energy minimization problem*. Although this problem is NP-hard, in general, a number of efficient approximative solvers exist, see [52] for a recent review.

4.4.4. POSE ESTIMATION AS ENERGY MINIMIZATION

Consider the following energy minimization problem:

- The set of nodes is the set of pixels of the input image, i.e., each graph node corresponds to a pixel. To be precise, we scale down our image by a factor of two for faster processing, i.e. each graph node corresponds to 2×2 pixels.

- Number of labels in every node is the same. The label set $L_u := \hat{L}_u \cup \{\mathrm{o}\}$ consists of two parts, a subset $\hat{L}_u$ of correspondence proposals and a special label o. In total, each node is assigned 13 labels: The forest $\mathcal{T}$ provides 3 candidates for object coordinates in each pixel, 2×2 pixels result in 12 labels, and the last label is the "outlier".

  Each label from the subset $\hat{L}_u$ corresponds a 3D coordinate on the object. Therefore, we will associate such labels l_u with 3D vectors and assume vector operations to be well-defined for them. Unary costs $\theta_u(l_u)$ for these labels are set to $(1 - p_c(u))\alpha$, where $p_c(u)$ is the object probability prediction provided by the random forest as defined in Section 2.2.2 and α is a hyper-parameter of our method. We will call the labels from $\hat{L}_u$ *inlier labels* or simply *inlier*.

  The special label o denotes a situation in which the corresponding node does not belong to the object, or none of the labels in $\hat{L}_u$ predicts a correct object

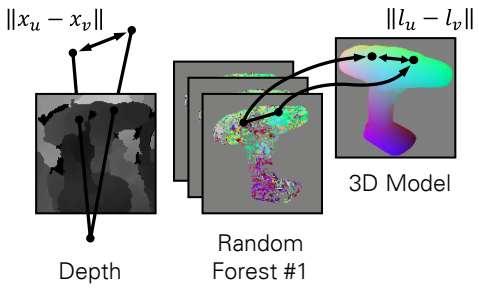


Figure 4.3.: **Binary Potentials.** Visualization of our Binary Potentials as defined in Equation 4.7.

coordinate. We call o the "outlier label". Unary costs for the outlier labels are: $\theta_u(\mathrm{o}) = \frac{\sum p_c(u)\alpha}{12}$, $u \in V$.

Let us define pose-consistent pixels. If a node, comprising of 2×2 pixels, is an inlier then the pixel with the respective label is defined as pose-consistent. The remaining three pixels are not pose-consistent and are ignored in the hypotheses selection stage. Also all pixels for which the node has an outlier label are not pose-consistent.

- Let x_u and x_v be 3D points in the camera coordinate system, corresponding to the nodes u and v in the scene (see Figure 4.4. For any two inlier labels $l_u \in \hat{L}_u$ and $l_v \in \hat{L}_v$ we assign the pairwise costs as follows

$$\theta_{uv}(l_u, l_v) = \begin{cases} \Big| \|l_u - l_v\| - \|x_u - x_v\| \Big|, & \|x_u - x_v\| \leq D \\ \infty, & \text{otherwise.} \end{cases} \tag{4.7}$$

  That is, $\theta_{uv}(l_u, l_v)$ is equal to the absolute difference of distances between points l_u, l_v on the object and x_u, x_v in the scene (see Figure 4.3) if the latter difference does not exceed the object size D.

  Additionally, we define $\theta_{uv}(l_u, \mathrm{o}) = \theta_{uv}(\mathrm{o}, l_v) = \gamma$ for $l_u \in L_u$, $l_v \in L_v$. Here γ is another hyper-parameter of our method. A sensible setting is $\gamma = 0$, however, we will choose $\gamma > 0$ in parts of the optimization (see details below). We also assign $\theta_{uv}(\mathrm{o}, \mathrm{o}) = 0$, for all $\{u, v\} \in E$.

- The graph G is fully-connected, i.e., any two nodes $u, v \in V$ are connected by an edge $\{u, v\} \in E$.

Given a labeling $l \in \mathbb{L}$ we will speak about *inlier* and *outlier* nodes as those labeled with inlier or outlier labels, respectively.

The energy of any labeling is a sum of (i) the total unary costs for inlier labels, (ii) total geometrical penalty of the inlier labels, and (iii) total cost for the outlier labels. A labeling with the minimal energy corresponds to a geometrically consistent subset

of coordinate correspondences with a certain confidence for the local classifiers. We believe, there are such hyper-parameter settings that these coordinates would provide approximately correct object poses.

Why a fully-connected graph? At the first glance, one could reasonably simplify the energy minimization problem described the above by considering a sparse, e.g. grid-structured graph. In this case the pairwise costs would control not all pairs of inlier labels, but only a subset of them, which may seem to be enough for a selection of inliers defining good quality correspondences. Unfortunately, such a simplification has a serious drawback, nicely described in [5]: As soon as the graph is not fully connected, it tends to select an optimal labeling, which contains separated "islands" of inlier nodes, connecting to other "inlier-islands" only via outlier nodes. Such a labeling may contain geometrically independent subsets of inlier labels, which may "hallucinate" the object in different places of the image. Moreover, from our experience many of such "islands" contain less than three nodes, which increases the probability for pairwise geometrical costs to be low just by chance.

Concerning energy minimization. Our graph contains 320×240 nodes which corresponds to the size of our discretized input image. Solving an energy minimization problem on such a fully-connected graph, even approximately, is in general infeasible if Gaussian potentials (like e.g. [63]) cannot be applied. Therefore, we suggest *a problem-specific*, but *very efficient* two-stage procedure for generating approximative solutions of the considered problem. In a first stage (Section 4.4.5) we reduce the size of the optimization problem, in the second (Section 4.4.6) we generate solution candidates.

4.4.5. STAGE ONE: PROBLEM SIZE REDUCTION

Despite what is discussed above about having a fully-connected graph, we used a sparse graphical model to reduce the number of possible correspondence candidates. An optimal labeling of this sparse model provides us with a set of inlier nodes, which hopefully contain the true inliers. On the second stage of our optimization procedure, described below, we build several fully-connected graphs from these nodes. For the sparse graph we use the following neighborhood structure: we connect each node to the 48 closest nodes and exclude the closest 8 since we believe that the distance measure between them is very noisy. We assign a positive value to the parameter γ penalizing transitions between inlier and outlier labels. This decreases the number of "inlier islands" by increasing the cost of the transition. We approximately solved this sparse problem with the TRW-S algorithm [61], which we run for 10 iterations. We found the recent implementation [106] of this algorithm to be up to 8 times faster than the original one [61] for our setting.

4.4.6. STAGE TWO: GENERATION OF SOLUTION CANDIDATES

Fully-Connected Graphical Model. As mentioned above, in the second stage we consider *a fully-connected* graphical model, where the node set contains only inlier nodes from the solution of the sparse problem. Moreover, to further reduce the problem size, we reduce the label set in each node to only two labels $L_u := \{0, 1\}$, where

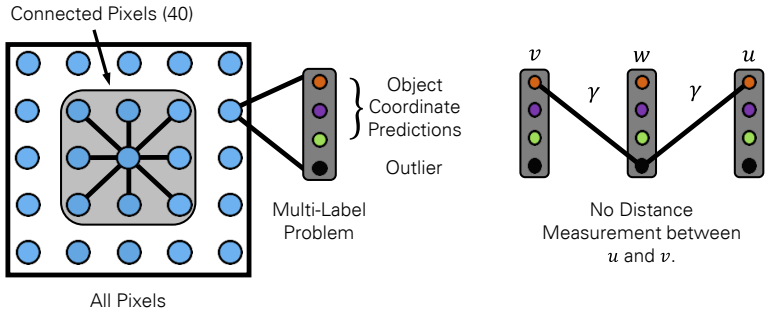


Figure 4.4.: **Illustrating Optimization Stage One. Left:**The graphical model is operating on the image with the set of nodes being the set of pixels (for the ease of visualization we did not apply pixel subsampling in this figure). Possible labels for each pixel are therefore the three object coordinates predicted by the random forest and an outlier label. The nodes within the graph are connected sparsely. **Right:** The geometric check is not applied between the nodes v and u since the node w connection the two is marked as an outlier.

the label 0 corresponds to an outlier and the label 1 corresponds to the label associated with the node in the solution of the sparse problem. The unary and pairwise costs are assigned as before, but the hyper-parameters α, β and γ are different. In particular $\gamma = 0$ since there is no reason to penalize transitions between inlier and outlier on this stage. Further, we will refer to $(G, \mathbb{L}, \theta)$ defined above, as to master (**fully-connected**) model F.

Although such problems usually have a much smaller size (the solution of the sparse problem typically contains 20 to 500 inliers) our requirements to a potential solver are much higher at this stage. Whereas in the first stage we require only that the set of inlier nodes contains enough of correct correspondences, the inliers obtained on the second stage must be *all* correct (have small geometrical error). Incorrect correspondences may deteriorate the final pose estimation accuracy. Therefore the quality of the solution becomes critical on this stage. Although problems of this size are often feasible for exact solvers, obtaining an exact solution may take multiple minutes or even hours. Therefore, we stick to the methods delivering only *a part of an optimal solution (partial optimal labeling)*, but being able to do this in a fraction of seconds, or seconds, depending on the problem size. Indeed, it is sufficient to have only three inlier to estimate the object pose.

Partial Labeling. A partial labeling can be understood as a vector $l \in \{0, 1, ?\}^{|V|}$ with only a subset $V' \subset V$ of coordinates assigned a value 0 or 1. The rest of coordinates take a special value $?$ = "unlabeled". The partial labeling is called *partial optimal labeling*, if there exists *an optimal* labeling $l^* \in \mathbb{L}$ such that $l^*_u = l_u$ for all $u \in V'$.

There are a number of efficient approaches addressing partial optimality (obtaining

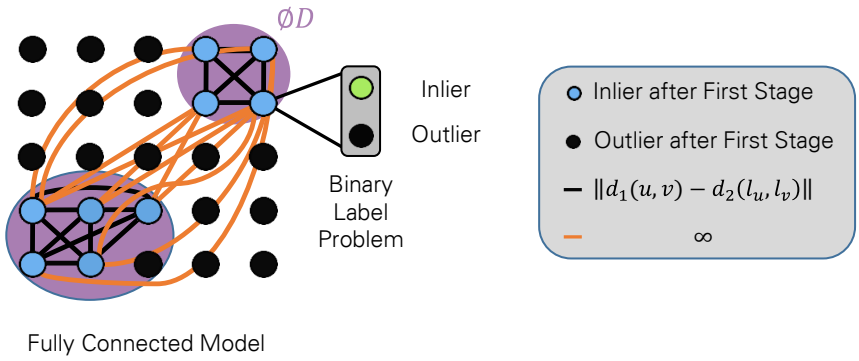


Figure 4.5.: **Illustrating Optimization Stage Two.** The blue pixels are all those pixels which were labeled as inliers, (potentially pose-consistent) in the first stage of the optimization. The first stage is opportunistic in the sense that wrong inliers may still be present. The goal of the second stage is to determine exactly the true inliers, from which we will determine the final pose. For this we have to solve the fully-connected graph shown, where each pixel has two labels, being an inlier (1) or outlier (0). Here the orange links mark pairwise terms which contain ∞ values. Unfortunately, state of the art solvers struggle with this problem, due to the presence of orange links. We approach this by solving two (in practice many more) submodels (two purple circles defined by the object diameter D) that contain no orange links. Each sub-problem produces a partial optimal solution $\{0, 1, ?\}$, where nodes that do not belong to the submodel are labeled 0. We can now guarantee that one of the partial optimal solution is the partial optimal solution of the full graph.

partial optimal labelings) for discrete graphical models for both multiple [114, 106] and two-label cases [62, 122]. We refer to [105] for an extensive overview. For problems with two labels the standard partial optimality method is QPBO [62], which we used in our experiments.

All partial optimality methods are based on sufficient optimality conditions, which have to be fulfilled for a partially optimal labeling. However, as it directly follows from [115, Prop.1], these conditions can hardly be fulfilled for label l_u in a node u, if for some neighboring node v: $\{u,v\} \in E$ the difference between the smallest pairwise potential "attached" to the label l_u, $\min_{l_v \in L_v} \theta_{uv}(l_u, l_v)$ and the largest one $\max_{l_v \in L_v} \theta_{uv}(l_u, l_v)$ is very large. In our setting this is the case, e.g. , if for two nodes u and v (connected by an edge as any pair in a fully-connected graph) it holds $\|x_u - x_v\| > D$, see (4.7). Existence of such infinite costs leads to deterioration of the QPBO result: in many cases the returned partial labeling contains less than three labeled nodes, which is not sufficient for pose estimation.

To deal with this issue, we propose a novel method to find *multiple* partial labelings: We consider a set of induced submodels (see Definition 1 below) and find a partial optimal solution for each of them. We guarantee, however, that *at least* one of these partial labelings is a partial optimal one *for the whole graphical model* and not only for its submodel. Considering submodels allows to significantly reduce the number of node pairs $\{u,v\}$ with $\theta_{uv}(1,1) = \infty$. In its turn, it leads to many more nodes being marked as partially optimal by QPBO and therefore, provides a basis for a high quality pose reconstruction (see Figure 4.5).

The theoretical background for the method is provided in the following subsection.

4.4.7. ON OPTIMALITY OF SUBPROBLEM SOLUTIONS FOR BINARY ENERGY MINIMIZATION

Let $G = (V, E)$ be a graph and $V' \subset V$ be a subset of its nodes. A subgraph $G' = (V', E')$ is called *induced* w.r.t. V', if $E' = \{\{u,v\} \in E \colon u, v \in V'\}$ contains all edges of E connecting nodes within V'.

Definition 1. *Let $M = (G, \mathbb{L}, \theta)$ be a graphical model with $G = (V, E)$ and $\mathbb{L} = \prod_{u \in V} L_u$. A graphical model $M' = (G', \mathbb{L}', \theta')$ is called* induced *w.r.t. $V' \subseteq V$ if*

- *G' is an induced subgraph of G w.r.t. V'.*
- *$\mathbb{L}' = \prod_{u \in V'} L_u$.*
- *$\theta'_u = \theta_u$ for $u \in V'$ and $\theta'_{uv} = \theta_{uv}$ for $\{u,v\} \in E'$.*

Proposition 1. *Let $M = (G, \mathbb{L}, \theta)$ be a graphical model, with $G = (V, E)$, $\mathbb{L} = \{0,1\}^{|V|}$ and θ such that*

$$\theta_{uv}(0,1) = \theta_{uv}(1,0) = \theta_{uv}(0,0) = 0 \quad \forall \{u,v\} \in E\,. \tag{4.8}$$

Let $\hat{l} \in \mathbb{L}$ be an energy minimizer of M and $\hat{V} := \{u \in V \colon \hat{l}_u = 1\}$.
Let $M' = (G', \mathbb{L}', \theta')$ be an induced model w.r.t. some $V' \supseteq \hat{V}$ and l' be an energy

minimizer of M'. Then there exists a minimizer l^* of energy of M, such that $l'_u = l^*_u$ for all $u \in V'$.

Proof. $E_V(\hat{l}) = E_{V'}(\hat{x}_{V'}) + E_{V\setminus V'}(\hat{x}_{V\setminus V'}) \geq E(l') + E_{V\setminus V'}(\overline{0})$. Since $x_{V\setminus V'} = \overline{0}$ due to (4.8), the equality holds. The inequality holds by definition of l'. Let us consider the labeling $l^* := (l', \overline{0})$ constructed by concatenation of l' on V' and $\overline{0}$ on $V\setminus V'$. Its energy is equal to the right-hand-side of the expression, due to (4.8). Since $\hat{l}$ is an optimal labeling, the inequality holds as equality and the labeling l^* is optimal as well. It finalizes the proof. □

Corollary 1. *Let under condition of Proposition 1 l' be a partial optimal labeling for M'. Then it is partial optimal for M.*

Note, since pairwise costs of *any* two-label (pairwise) graphical model can be easily transformed to the form (4.7), see e.g. [62], Proposition 1 is generally applicable to all such models.

4.4.8. OBTAINING CANDIDATES FOR PARTIAL OPTIMAL LABELING

To be able to use Proposition 1 we need a way to characterize possible optimal labelings for the master model F (defined in Section 4.4.6) to be able to generate possible sets V' containing all inlier nodes of an optimal labeling. Indeed, this characterization is provided by the following proposition:

Proposition 2. *Let l^* be an optimal solution to the fully-connected problem described above. Then, for any two inlier nodes u and v, $l^*_u = l^*_v = 1$, it holds $\|x_u - x_v\| \leq D$ or, in other words, $\theta_{uv}(l^*_u, l^*_v) < \infty$.*

This proposition has a trivial proof: as soon as there is a labeling with a finite energy (e.g. $l_u = 0$ for all $u \in V$), an optimal labeling can not have an infinite one.

An implication of the proposition is quite clear from the applied point of view: all inlier nodes must be placed within a circle with a diameter equal to the maximal linear size of the object. Combining this observation with Proposition 1, we will generate a set of submodels, which contain all possible subsets of nodes satisfying the above condition.

A simple, yet inefficient way to generate all such submodels, is to go over all nodes u of the graph G and construct a subproblem M_u induced by nodes, which are placed at most at the distance D of u. A disadvantage of this method is that one gets as many as $|V|$ subproblems, which leads to the increased runtime and too many almost equal submodels. Instead, we consider all connected inlier components obtained on the first stage as a result of the problem reduction. We remove all components with the size less than three, because, as we found experimentally, they mostly represent only noise. We enumerate all components, i.e., assign a serial number to each. For each component f we build a fully-connected submodel, which includes itself and all components with bigger serial number within the distance D from all nodes of f. Such an approach usually leads to at most 20 submodels and most of them get more than three partial optimal labels by QPBO.

Ignoring the heuristic removal of the components with the size less than three, such a procedure is guaranteed to provide a partial optimal solution of the whole problem, independent of the selected ordering of the components. Indeed, let an optimal labeling include inliers from $m > 1$ components. Then select the component with the smallest index out of these m ones. By construction, the corresponding submodel will contain all the m components (since they all lie within the distance D and have larger indices) and therefore all the inliers of an optimal solution.

4.4.9. REFINEMENT AND HYPOTHESIS SCORING

The output of the optimization of the graphical model is a collection of pose-consistent pixels where each of those pixels has a unique object coordinate. The collection is clustered into sets. In the example in Figure 4.2(c) there are two sets (red, green). Each set provides one pose hypothesis. These pose hypotheses are refined and scored using our ICP-variant. In order to be robust to occlusion we only take the pose-consistent pixels within the ICP [6, 99] for fitting the 3D model.

4.5. EXPERIMENTS

We evaluated our method on a publicly available dataset. We will first introduce the dataset and then the evaluation protocol (Section 4.5.1). After that, we quantitatively compare our work with three competitors, and also present qualitative results (Section 4.5.2).

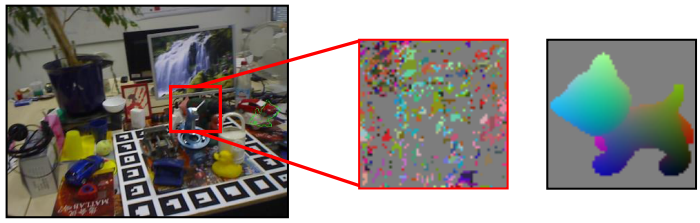


Figure 4.6.: **Failure Case.** We use the random forest from [8] that was trained on image patches of non-occluded objects. Hence they can only handle a moderate level of occlusion. In case of strong occlusion they fail to predict precise object coordinates. In the illustrated example, a wrong pose is predicted (green silhouette) and the object coordinates are also wrong (see zoom). In future work, this problem can be mitigated for instance by training on image patches that contain occlusions.

4.5.1. DATASET

To evaluate our method, we use the publicly available dataset of Brachmann et al. [8], known as "Occluded Object Dataset"[3]. This dataset was presented in [8] and is an extension of [37]. They annotated the ground truth pose for 8 objects in 1214 images with various degrees of object occlusions.

To evaluate our method we use the criteria from [37]. This means that we measure the percentage of correctly estimated poses for each object. To determine the quality of an estimated pose we calculate the average distance of each point within the object with respect to the estimated pose and the ground truth pose. The pose is accepted if the average distance is below 10% of the object diameter.

We annotated an additional image sequence (1235 images) of the dataset [37] containing 6 objects and used it as a validation dataset for the parameter setting of the graphical model. Further details of the dataset are presented in Section A.2.1. The final set of parameters for stage one is $\alpha = 0.21, \beta = 23.1, \gamma = 0.0048$ and stage two is $\alpha = 0.2, \beta = 2.0, \gamma = 0.0$.

Object	Method			
	Our method	Hinterstoisser et al. [38]	Krull et al. [64]	Brachmann et al. [8]
Ape	80.7%	**81.4%**	68.0%	53.1%
Can	88.5%	**94.7%**	87.9%	79.9%
Cat	**57.8%**	55.2%	50.6%	28.2%
Driller	**94.7%**	86.0%	91.2%	82.%
Duck	74.4%	**79.7%**	64.7%	64.3%
Eggbox	47.6%	**65.6%***	41.5%	9.0%
Glue	**73.8%**	52.1%	65.3%	44.5%
Hole Puncher	**96.3%**	95.5%	92.9%	91.6%
Average	76.7%	76.3%	70.3%	56.6%

Table 4.2.: Quantitative comparison of [8], [64], [38] and our approach for all objects in the challenging "Occluded Object Dataset". *The number for the Eggbox differs from [38] since they did not consider all images of the sequence (private e-mail exchange with the authors).

4.5.2. RESULTS

In the following we compare to the methods of Brachmann et al. [8], Krull et al. [64] and to the recently published state-of-the-art method of Hinterstoisser et al. [38]. Results are shown in Table 4.2. We achieve an average accuracy of 76.7% over all objects, which is 0.4% better than the current state-of-the-art method of Hinterstoisser

[3] http://cvlab-dresden.de/iccv2015-occlusion-challenge/

et al. [38]. With respect to individual objects our method performs best on four objects and [38] on the other four. In comparison with [8] and [64] we achieve an improvement of 20.1% and 6.4% respectively. Since these two methods use the same random forest, as we do, the benefits of using global reasoning can be seen. See Figure 4.7 for qualitative results.

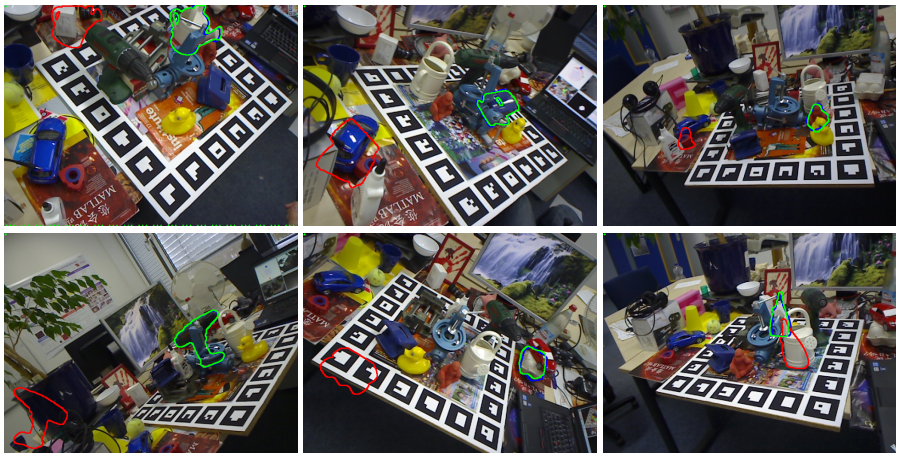


Figure 4.7.: **Results - Global Hypothesis Generation.** Qualitative results on the "Occluded Object Dataset" [8]. Results of our method are depicted as green silhouettes, the ground truth pose is shown as a blue silhouette and results of the method by Krull et al. [64] are shown as red silhouettes. Note, since these results shows correct poses of our method the green silhouette is on top of the blue one.

4.6. SUMMARY

In this chapter, we presented a system addressing the task of 6D pose estimation under severe occlusion. We introduced a novel, global geometry check in form of a fully connected CRF. Since this direct optimization on the CRF is hardly feasible, we present an efficient two-step optimization procedure, with some guarantees on optimality.

5. DISCUSSION

Contents

5.1. Accuracy 76
5.2. Scalability 77
5.3. Global Hypothesis Generation as an End-to-End Pipeline . . . 78
5.4. 6D Pose Challenge 79

In this thesis, we presented three different systems approaching the task of pose estimation of object instances. We were focusing on the hypothesis generation process within the pose estimation system. In particular, we employed task knowledge to improve both the accuracy as well as the efficiency of the hypothesis generation. Our systems were applied within indoor (domestic) and outdoor (airport apron) environments showing that they are applicable in realistic scenarios. As input data to our systems we used both RGB-D and depth data only. We were able to show that our methods are not tied to a specific type of sensors as we used both Lidar-based and structured light-based sensors. Although we addressed different challenges, the task of pose estimation is still far from being solved. In the following, we will discuss the advantages as well as the limitations of the proposed systems.

5.1. ACCURACY

Our pose estimation systems have shown their ability to reliably estimate poses of rigid object instances and articulated objects given a single image as input. They are not restricted to a specific sensor type, can handle textured as well as texture-less objects and have shown to be robust to self occlusion and occlusion by unknown objects. We conducted pose estimation of large objects (see Chapter 2), furniture (see Chapter 3) and small objects (see Chapter 4). The quality of the pose estimates provided by our systems allows them to be used for robot grasping and augmented reality applications. Additionally, the pose accuracy provided by our system for aircraft objects fulfill the standards defined by the International Civil Aviation Organization making them applicable for the apron monitoring task. At the time of publication our systems defined the state-of-the-art.

Our systems are able to handle various object scales and object types but there is still a variety of challenges remaining. Transparent and reflective materials, for example, cause both Lidar-based and structured light-based sensors to fail in providing reliable depth measurements. This is similar for RGB sensors since the object appearance is heavily influenced by objects being visible through a transparent object or reflections of the surrounding environment for reflective objects. Object contours are mostly unaffected by those effects and Phillips et al. [91] successfully employed this cue for pose estimation of transparent objects. Another object type that causes difficulties are thin objects like cutlery and office requisites since they are often not rich in textural features and depth sensors struggle to provide reliable measurements.

Measurements provided by depth sensors are not affected by changes in illumination. The object appearance in RGB images, in contrast, is heavily affected by lighting changes. However, RGB images contain rich information which can be utilized to resolve ambiguities of object shape, e.g. the pose of a book can only be exactly determined if the front cover can be distinguished from the back cover. Robustness towards lighting changes can be achieved using features being invariant to such changes. Brachmann et al. [8] used training data showing the objects under varying lighting conditions to learn such features. A different approach to gain robustness towards illumination changes was introduced by Horn [43]. He proposed a system estimating the lighting conditions present in the image. This process is also called inverse rendering as is decomposes the image into different intrinsic layers describing geometry, lighting and object reflectance properties. This information enables the calculation of the true object texture.

We approached pose estimation of articulated objects. Such objects are assemblies of multiple parts that are structured by inter-part connections. While those articulated objects are deformable, we only considered assemblies of rigid parts. Many objects, e.g. clothing or toys, show additional degrees of freedom where the deformation is not determined by a clear structure. While such deformations can be addressed by introducing additional deformation parameters, the number of parameters needed can grow fast making pose estimation using random samping technqiues inefficient. However, pose estimation of objects featuring deformation have been proposed by Taylor et al. [117] for human bodies as well as by Sharp et al. [104] for hands using optimization techniques to determine the high dimensional object pose efficiently.

Deformation properties are also evident when considering pose estimation of object classes, e.g. cars or bicycles. While we only approached pose estimation of object instances the methods of Winn et al. [124] and Hoiem et al. [42] both used the concept of discrete object coordinates and a MRF-inference process for the hypothesis generation which is similar to our approach presented in Chapter 4.

5.2. SCALABILITY

The proposed systems employing random sampling to generate pose hypotheses (see Chapter 2 and 3) require less than one second to provide pose estimates for one image. Our system using global reasoning (see Chapter 4) to generate hypotheses provides results in one to three seconds per image. While such run times are sufficient for pose

estimation of static objects, e.g. stationary object grasping, for applications scenarios considering moving objects, e.g. augmented reality, run times of 100 milliseconds and lower are required. Object tracking apporaches facilitate real time pose estimation by utilizing information from previous time steps. Krull et al. [66], for example extended the system of [8] to conduct object tracking. They used the concept of object coordinate regression within an 6D object tracking system to gain robustness towards fast object motion and occlusion. They were able to estimate object poses at frame rates of 20 images per second. Our systems proposed in Chapter 2 and 3 can also be extended in a similar manner.

All our system use random forests to solve the correspondence problem. This part of the pipeline is therefore scaling logarithmically with the number of objects. Since we assume that the object of interest is present in the image and therefore omit object detection, increasing numbers of objects do not influence the run time of our methods. However, our random sampling based systems can be extended to efficiently conduct pose estimation of multiple objects which has been shown by Brachmann et al. [10], who proposed a hypothesis generation process scaling sublinear with the number of objects.

When considering real world applications like robot bin picking in a warehouse the number of objects might go into millions or even tens of millions. In such a scenario logarithmic scaling is not sufficient. This issue can however be solved by employing a two stage process were the object is detected first and pose estimation is conducted in the second stage.

We considered articulated objects with up to four rigid parts. Using a kinematic chain representation enabled us to generate pose hypotheses for an object consisting of k parts using k correspondences. The hypothesis generation is therefore scaling linearly with the number of parts within the articulated object.

We only considered articulated objects with one degree of freedom joints. This is however not a limitation of idea in general. Our system can be extended to consider two degrees and even three degrees of freedom joints. This would however imply, that we need to sample two, respectively three more correspondences to determine the articulated pose. The benefits of employing the kinematic chain structure in the hypothesis generation would still persist.

5.3. GLOBAL HYPOTHESIS GENERATION AS AN END-TO-END PIPELINE

In Chapter 4, we presented a system addressing the challenges of object occlusion. We formalized the hypothesis generation process as an energy minimization process which increased the accuracy of pose estimation under occlusion.

There are several parts within our pose estimation pipeline where machine learning can be applied. The correspondence estimation is the first part of this pipeline and we applied the concept of object coordinate regression, which employs random forests, to solve this task. Brachmann et al. [9] proposed to use a CNN to predict image-to-object correspondences which improved the quality of the predictions and the quality of the estimated poses.

Machine learning techniques can also be applied in the second part of the pipeline, the hypothesis generation. There is a large variety of works approaching the task of learning parameterizations of graphical models, e.g. [97, 84, 48]. We did parameterize the cost functions of our graphical model, however, we did not learn those parameters, but employed an elaborate grid search process to find good values.

There are also opportunities to apply machine learning in the refinement and scoring steps of our pipeline. Krull et al. [64, 65] proposed systems which approached learning of scoring and refinement functions for 6D pose estimation. Brachmann et al. [9] also learned a scoring function for the task of camera re-localization. In our work, we used an ICP algorithm which also contains multiple free parameters which were adjusted manually.

As discussed, there are multiple opportunities for learning techniques to be applied to the individual parts of the pipeline. However, as shown by Brachmann et al. [9], it is sensible to perform learning in an end-to-end fashion to enable back-propagation of the errors through the whole system. This is also possible for our proposed pipeline, but there are several challenges which need to be addressed.

We used synthetically generated images of the objects to train the random forest predicting image-to-object correspondences. This is preferable, since such images are easy to create and provide perfect ground truth information. There is however a large shift between the images used to train the forest and the images captured by the sensor during test time. While the random forest is able to generate features being robust to the domain shift, we do not expect the same behavior for CNNs. The method by Kehl et al. [56] employs a CNN for the task of pose estimation and also uses synthetic data for training. They did provide good results for pose estimation of unoccluded objects, but they did not show results for the pose estimation under occlusion. It is therefore an interesting question for future works to enable CNNs to provide high quality predictions when trained on synthetic data.

Multiple systems combining CRFs and CNNs within a end-to-end trainable system have recently been proposed [103, 128]. They use the benefits of CNNs, which are able to learn large sets of parameters efficiently, and the benefits of CRFs, which allow to incorporate prior knowledge. Those systems assume that the potential functions of the CRF are gaussian which renders them incompatible to our system. The works of Kirillov et al. [60] and Chen et al. [16] enabled end-to-end learning without the restriction to gaussian potentials. Incorporating their findings into our pose estimation pipeline is a promising direction for future work.

5.4. 6D POSE CHALLENGE

The dataset of Hinterstoisser et al. [37] marked a milestone in the field of 6D pose estimation, since it enabled researchers to benchmark their efforts. Since then, many other datasets addressing the task of pose estimation have been published. They cover a variety of tasks, e.g. showing multiple objects [39] or multiple instances [118], bin picking [94], and robot part assembly [39, 21]. Furthermore, they address multiple challenges of pose estimation, e.g. lighting changes, occlusion and clutter [8], reflective materials [21] and ambiguities in shape and texture [39].

Many methods approaching the task of pose estimation and object detection have been applied to the aforementioned datasets. Comparing the results of those methods is cumbersome since only rarely all datasets are considered by a method and different metrics are used to evaluate the accuracy. Evaluating a method on all available datasets is time consuming since they do usually not follow a standardized format. They differ in the format of ground truth annotations, the format of the 3D object model and the image data. It is therefore difficult to determine the state-of-the-art of pose estimation.

Several challenges and benchmarks have been proposed in the domain of computer vision in the past, e.g. the Middlebury dataset [101] for stereo vision, the Pascal VOC challenge [26] for object detection, the Kitty benchmark [32] for 3D object detection and tracking, and the Microsoft COCO challenge [69] for image segmentation. Those challenges enabled researchers to compare their methods to a wide field of competitors which led to an advance of the state-of-the-art.

Such a standardized benchmark is not available for the task of pose estimation. However, it is not necessary to acquire new data since many challenges have already been addressed. Therefore, the effort which needs to be invested is the combination and unification of datasets already being available.

6. CONCLUSION

Knowing the pose of an object is beneficial in many application scenarios. There is a large variety of applications depending on object pose information and their number is growing continuously. While many challenges of pose estimation have been approached in the past, the task is still far from being solved. This makes pose estimation an important and exciting field for future works.

In this thesis, we have shown that the incorporation of geometric task knowledge is beneficial for solving the challenges arising with the task of pose estimation. The proposed systems are robust, versatile, scalable, and operate fast enough to be useful for various applications. In the previous chapter we have shown that there are many exciting open research questions in the field of pose estimation.

A. APPENDIX

Contents

A.1. Abbreviations 84
A.2. Datasets 85
A.2.1. Occlusion Datasets 85
A.2.2. Articulated Objects Dataset (Our) 86
A.3. Derivation for the Estimation of Articulation Parameters. . . . 88

A.1. ABBREVIATIONS

AR Augmented Reality

CNN Convolutional Neural Network

CRF Conditional Random Field

DOF Degree of Freedom

DPM Deformable Parts Model

FPFH Fast Point Feature Histogram

HOG Histogram of oriented gradients

ICP Iterative Closest Point

Lidar Light Detection and Ranging

MRF Markov Random Field

NP Non-deterministic Polynomial-time

QPBO Quadratic Pseudo-Boolean

RANSAC Random Sample Consensus

RGB Image featuring red, green and blue color channels

RGB-D Image featuring color channels and an additional depth channel

SIFT Scale-Invariant Feature Transform

SLAM Simultaneous Localization and Mapping

SURF Speeded-up Robust Features

TRWS Tree-reweighted Message Passing

A.2. DATASETS

As discussed in Section 5.4 there is a large variety of available datasets addressing the task of object pose estimation. Hinterstoisser et al. [37] were the first to compile a dataset providing accurate 6D pose annotations. They captured images showing the object under varying poses in a cluttered environment. The objects were however not substantially occluded. To address object occlusion Brachmann et al. [8] extended the dataset of [37] with additional ground truth annotations. We further created a second extension of the Hinterstoisser dataset [37]. We will provide details of the two extensions in Section A.2.1.

To evaluate our system on articulated pose estimation, we compiled a dataset showing several objects in different articulation states. We will discuss this dataset in Section A.2.2.

A.2.1. OCCLUSION DATASETS

The dataset of Hinterstoisser et al. [37] contained 15 objects. An image sequence was captured for each object and the ground truth was provided as the rotation and the translation of the object relative to the camera. Each individual sequence contained approximately 1200 images showing the object under varying camera viewpoints being distributed over the upper object hemisphere. The dataset contained digital 3D models for 13 objects. They were reconstructed using images captured with a low-cost sensor similar to the Microsoft Kinect.

OCCLUSION DATASET (BRACHMANN ET AL. [8])

While the objects in the dataset of Hinterstoisser et al. [37] were positioned in a cluttered environment, they are never substantially occluded. This inspired Brachmann et al. [8] to extend the ground truth information to all known objects within one image sequence, namely the "Bench Vise" image sequence. They annotated ground truth poses for eight additional objects resulting in approximately 8500 additional ground truth annotations (see Table A.2.1 for statistics). Figure A.1 shows one image of this sequence on the left, and 3D models of the objects for which additional ground truth was created on the right.

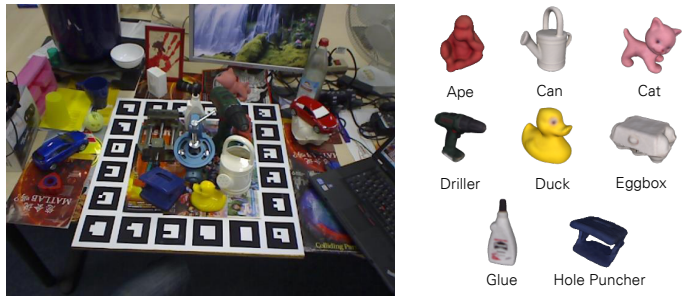


Figure A.1.: **Occlusion Extension of [8]. Left:** An image of the "Bench Vise" image sequence were the ground truth information was extended to all known objects. **Right:** Objects were additional ground truth was created, from left to right, top to bottom: Ape, Can, Cat, Driller, Duck, Eggbox, Glue, Hole Puncher.

OCCLUSION DATASET (OUR)

We further annotated a second image sequence of the Hinterstoisser dataset [37], namely the "Ape" image sequence. We provide ground truth information for six additional objects resulting in approximately 7000 additional ground truth annotations (see Table A.2.1 for statistics). Figure A.2 shows one image of this sequence on the left and the objects for which ground truth annotations were annotated on the right.

A.2.2. ARTICULATED OBJECTS DATASET (OUR)

While several datasets addressing the task of 6D pose estimation are publicly available, there is no dataset which addresses pose estimation of articulated objects consisting of rigid parts. We therefore created a new dataset featuring four objects with two, three and four rigid parts. The objects can be described as kinematic chains, where a connection between two parts is either a rotation around one axis or a translation along one axis. We captured two image sequences for each object resulting in a total number of 7047 images ((see Table A.2.2 for statistics)). While the articulation changes between the sequences, the objects are static throughout one sequence. We created the digital 3D object models shown in Figure A.3 using computer aided design.

	Object dimensions	# Frames	Occlusion		
			0%-33%	33%-66%	66%-100%
Ape	8cm × 9cm × 8cm	1170	1051	117	2
Can	18cm × 19cm × 10cm	1207	1119	84	4
Cat	13cm × 12cm × 7cm	1187	1097	72	18
Driller	8cm × 21cm × 23cm	1214	1100	102	12
Duck	8cm × 9cm × 10cm	1143	1069	74	0
Eggbox	11cm × 7cm × 15cm	1175	990	132	53
Glue	8cm × 17cm × 4cm	901	836	54	11
Hole Puncher	13cm × 10cm × 11cm	1210	1185	22	3

Table A.1.: **Statstics of the Extension of [37] by [8].** We provide the object dimensions and the number of additionally annotated frames. Furthermore, we state the number of images depending on the ratio of occlusion. The occlusion was estimated by rendering the object models under the ground truth pose and comparing the synthetic depth image to the depth image captured by the sensor.

Object	Object dimensions	# Frames	Occlusion		
			0%-33%	33%-66%	66%-100%
Can	18cm × 19cm × 10cm	1235	1182	34	19
Cat	13cm × 12cm × 7cm	1235	1128	7	0
Duck	8cm × 9cm × 10cm	1207	1200	7	0
Eggbox	11cm × 7cm × 15cm	1160	1027	88	45
Glue	8cm × 17cm × 4cm	1235	1212	22	1
Hole Puncher	13cm × 10cm × 11cm	1235	1194	32	9

Table A.2.: **Statstics of our Extension to [37].**We provide the object dimensions and the number of additionally annotated frames. Furthermore, we state the number of images depending on the ratio of occlusion. The occlusion was estimated by rendering the object models under the ground truth pose and comparing the synthetic depth image to the depth image captured by the sensor.

Figure A.2.: **Occlusion Extension (Our). Left:** An image of the "Ape" image sequence were ground truth information was extended to all objects contained within the dataset. **Right:** Objects were additional ground truth was created, from left to right, top to bottom: Can, Cat, Duck, Eggbox, Glue, Hole Puncher.

A.3. DERIVATION FOR THE ESTIMATION OF ARTICULATION PARAMETERS.

Here we provide the derivation for Equation 3.2 and Equation 3.3 used to find articulation parameters from two point correspondences in Section 3.3.3.

REVOLUTE JOINTS

We will first consider the case of revolute joints. We show the derivation for a rotation around the x-axis. The task is to derive the angle at the joint from two correspondences $(\mathbf{x}(i_1), \mathbf{y}_k(i_1))$ and $(\mathbf{x}(i_2), \mathbf{y}_{k+1}(i_2))$, with $\mathbf{y}_k(i) = (\mathsf{x}_k, \mathsf{y}_k, \mathsf{z}_k)^\top$. We abbreviate the squared distance between the two points in camera space as $d_\mathbf{x} = \|\mathbf{x}(i_1) - \mathbf{x}(i_2)\|^2$. We start with Equation 3.1 and solve for θ.

Figure A.3.: **Articulated Objects Dataset.** The row top shows the objects contained within the dataset. From left to right: Laptop, Cabinet, Cupboard, Toy Train. Images in the middle and bottom row shown one image for each individual object sequence.

$$d_{\mathsf{x}} = \|\mathsf{y}_k(i_1) - A_k(\theta_k)\mathsf{y}_{k+1}(i_2)\|^2, \text{ with } A_k(\theta_k) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_k & -\sin\theta_k & 0 \\ 0 & \sin\theta_k & \cos\theta_k & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.1}$$

$$d_{\mathsf{x}} = (\mathsf{x}_k - \mathsf{x}_{k+1})^2 + (\mathsf{y}_k - (\cos(\theta_k)\mathsf{y}_{k+1} - \sin(\theta_k)\mathsf{z}_{k+1}))^2 + (\mathsf{z}_k - (\sin(\theta_k)\mathsf{y}_{k+1} + \cos(\theta_k)\mathsf{z}_{k+1}))^2 \tag{A.2}$$

$$\begin{aligned} d_{\mathsf{x}} = {} & (\mathsf{x}_k - \mathsf{x}_{k+1})^2 + \mathsf{y}_k^2 - 2(\mathsf{y}_k\cos(\theta_k)\mathsf{y}_{k+1} - \mathsf{y}_k\sin(\theta_k)\mathsf{z}_{k+1}) + (\cos(\theta_k)\mathsf{y}_{k+1} - \sin(\theta_k)\mathsf{z}_{k+1})^2 \\ & + \mathsf{z}_k^2 - 2(\mathsf{z}_k\sin(\theta_k)\mathsf{y}_{k+1} + \mathsf{z}_k\cos(\theta_k)\mathsf{z}_{k+1}) + (\sin(\theta_k)\mathsf{y}_{k+1} + \cos(\theta_k)\mathsf{z}_{k+1})^2 \end{aligned} \tag{A.3}$$

$$\begin{aligned} d_{\mathsf{x}} = {} & (\mathsf{x}_k - \mathsf{x}_{k+1})^2 + \mathsf{y}_k^2 + \mathsf{z}_k^2 \\ & + 2\sin(\theta_k)(\mathsf{y}_k\mathsf{z}_{k+1} - \mathsf{z}_k\mathsf{y}_{k+1}) + 2\cos(\theta_k)(-\mathsf{y}_k\mathsf{y}_{k+1} - \mathsf{z}_k\mathsf{z}_{k+1}) \\ & + 2\sin(\theta_k)\cos(\theta_k)(\mathsf{z}_{k+1}\mathsf{y}_{k+1} - \mathsf{z}_{k+1}\mathsf{y}_{k+1}) \\ & + \cos^2(\theta_k)(\mathsf{y}_{k+1}^2 + \mathsf{z}_{k+1}^2) + \sin^2(\theta_k)(\mathsf{y}_{k+1}^2 + \mathsf{z}_{k+1}^2) \end{aligned} \tag{A.4}$$

$$d_{\mathsf{x}} = (\mathsf{x}_k - \mathsf{x}_{k+1})^2 + \mathsf{y}_k^2 + \mathsf{z}_k^2 + \mathsf{y}_{k+1}^2 + \mathsf{z}_{k+1}^2 + a\sin(\theta_k) + b\cos(\theta_k), \tag{A.5}$$

where $a = 2(\mathsf{y}_k\mathsf{z}_{k+1} - \mathsf{z}_k\mathsf{y}_{k+1})$ and $b = -2(\mathsf{y}_k\mathsf{y}_{k+1} + \mathsf{z}_k\mathsf{z}_{k+1})$. It is known that

$$a\sin(\theta_k) + b\cos(\theta_k) = \sqrt{a^2 + b^2}\sin(\theta_k + \text{atan2}(b, a)). \tag{A.6}$$

	Parts	Dimensions	#Frames Sequence 1	#Frames Sequence 2
Laptop	Body	32cm 2cm 22cm	930	835
	Display	32cm × 2cm × 23cm		
Cabinet	Door	40cm × 49cm × 5cm	1108	1119
	Body	55cm × 75cm × 43cm		
	Drawer	40cm × 10cm × 39cm		
Cupboard	Body	134cm × 71cm × 40cm	901	901
	Drawer	42cm × 25cm × 39cm		
Toy train	Loco	10cm × 10cm × 15cm	847	404
	Waggon #1	10cm × 5cm × 16cm		
	Waggon #2	10cm × 9cm × 16cm		
	Waggon #3	10cm × 8cm × 16cm		

Table A.3.: **Statistics of the Articulated Object Dataset.** We provide the object dimensions for each part of the articulated object and the number of frames captured for each image sequence.

We can use supplementary Equation A.6 and continue

$$d_{\mathsf{x}} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{z}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_{k+1}^2 = \sqrt{a^2 + b^2}\sin(\theta_k + \mathsf{atan2}(b, a)) \quad \text{(A.7)}$$

$$\frac{d_{\mathsf{x}} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{z}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2 + b^2}} = \sin(\theta_k + \mathsf{atan2}(b, a)). \quad \text{(A.8)}$$

When we apply the asin function we have to consider the two possible results:

$$\mathsf{asin}\left(\frac{d_{\mathsf{x}} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{z}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2 + b^2}}\right) = \theta_k + \mathsf{atan2}(b, a) \quad \text{(A.9)}$$

and

$$\pi - \mathsf{asin}\left(\frac{d_{\mathsf{x}} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{z}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2 + b^2}}\right) = \theta_k + \mathsf{atan2}(b, a)), \quad \text{(A.10)}$$

which lead to the two solutions from Equation 3.2:

$$\theta_k^1 = \mathsf{asin}\left(\frac{d_{\mathsf{x}} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_k^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2 + b^2}}\right) - \mathsf{atan2}(b, a) \quad \text{(A.11)}$$

and

$$\theta_k^2 = \pi - \mathsf{asin}\left(\frac{d_{\mathsf{x}} - (\mathsf{x}_k - \mathsf{x}_{k+1})^2 - \mathsf{y}_k^2 - \mathsf{y}_{k+1}^2 - \mathsf{z}_k^2 - \mathsf{z}_{k+1}^2}{\sqrt{a^2 + b^2}}\right) - \mathsf{atan2}(b, a). \quad \text{(A.12)}$$

PRISMATIC JOINTS

We will now derive Equation 3.3, which addresses prismatic joints. We show the derivation for a translation along the x-axis. We start again with Equation 3.1 from Section 3.3.3:

$$d_{\mathsf{x}} = \|\mathbf{y}_k(i_1) - A_k(\theta_k)\mathbf{y}_{k+1}(i_2)\|^2, \text{ with } A_k(\theta_k) = \begin{pmatrix} 1 & 0 & 0 & \theta_k \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.13}$$

$$d_{\mathsf{x}} = (\mathsf{x}_k - (\mathsf{x}_{k+1} + \theta_k))^2 + (\mathsf{y}_k - \mathsf{y}_{k+1})^2 + (\mathsf{z}_k - \mathsf{z}_{k+1})^2 \tag{A.14}$$

$$d_{\mathsf{x}} = \mathsf{x}_k^2 - 2\mathsf{x}_k(\mathsf{x}_{k+1} + \theta) + (\mathsf{x}_{k+1} + \theta)^2 + (\mathsf{y}_k - \mathsf{y}_{k+1})^2 + (\mathsf{z}_k - \mathsf{z}_{k+1})^2 \tag{A.15}$$

$$d_{\mathsf{x}} = \mathsf{x}_k^2 - 2\mathsf{x}_k\mathsf{x}_{k+1} - 2\mathsf{x}_k\theta + \mathsf{x}_{k+1}^2 + 2\mathsf{x}_{k+1}\theta + \theta^2 + (\mathsf{y}_k - \mathsf{y}_{k+1})^2 + (\mathsf{z}_k - \mathsf{z}_{k+1})^2 \tag{A.16}$$

$$0 = \theta^2 + 2(\mathsf{x}_{k+1} - \mathsf{x}_k)\theta + \mathsf{x}_k^2 - 2\mathsf{x}_k\mathsf{x}_{k+1} + \mathsf{x}_{k+1}^2 + (\mathsf{y}_k - \mathsf{y}_{k+1})^2 + (\mathsf{z}_k - \mathsf{z}_{k+1})^2 - d_{\mathsf{x}}. \tag{A.17}$$

We can reformulate the equation as

$$0 = \theta^2 + p\theta + q, \tag{A.18}$$

with $p = 2(\mathsf{x}_{k+1} - \mathsf{x}_k)$ and $q = (\mathsf{x}_k - \mathsf{x}_{k+1})^2 + (\mathsf{y}_k - \mathsf{y}_{k+1})^2 + (\mathsf{z}_k - \mathsf{z}_{k+1})^2 - d_{\mathsf{x}}$. Solving supplemental Equation A.18 is a standard problem. The solutions are equivalent to Equation 3.3 in the paper:

$$\theta_k^1 = -\frac{p}{2} + \sqrt{\left(\frac{p}{2}\right)^2 - q} \tag{A.19}$$

and

$$\theta_k^2 = -\frac{p}{2} - \sqrt{\left(\frac{p}{2}\right)^2 - q}. \tag{A.20}$$

LIST OF FIGURES

1.1. **Pose Estimation Challenges**, Source Images: LINEMOD ACCV dataset (Author: S. Hinterstoisser, Published: 2012, CC BY 4.0 License), TUD-Light dataset (Author: S. Ihrke, Published: 2018) . . . 20
1.2. **Camera Re-localization and Articulated Objects.** Source Images: Microsoft 7-Scenes dataset (Authors: A. Criminisi, A. Fitzgibbon, J. Shotton, Published: 2013, MSR-LA License) . . . 21
1.3. **Multiple Objects and Multiple Instances.** Source Images: T-LESS dataset (Author: T. Hodan, Published: 2017, CC BY 4.0 License) . . . 22
1.4. **Pose Tracking.** Source Images: RGB-D Tracking dataset (Author: A. Krull, Published: 2014) . . . 22
1.5. **Applications: Monitoring and Robotics.** . . . 24
1.6. **Applications: Medicine and Augmented Reality.** Source Images: Endoscopic Vision Challenge: Instrument Segmentation and Tracking (Authors: Consortium for Open Medical Image Computing, Published: 2015) 25

2.1. **Decision Tree.** . . . 36
2.2. **Object Coordinate Regression.** . . . 37
2.3. **Training Data Generation.** . . . 40
2.4. **System Overview.** . . . 41
2.5. **Hypothesis Scoring.** . . . 42
2.6. **Results - Sampling-based Pose Estimation.** . . . 43

3.1. **Articulation Estimation.** . . . 48
3.2. **Comparison of Articulation Estimation.** . . . 54
3.3. **Results - Articulated Pose Estimation.** . . . 56

4.1. **Motivation.** Source Images: LINEMOD ACCV dataset (Author: S. Hinterstoisser, Published: 2012, CC BY 4.0 License) . . . 59
4.2. **Global Hypothesis Generation Pipeline.** Source Images: LINEMOD ACCV dataset (Author: S. Hinterstoisser, Published: 2012, CC BY 4.0 License) . . . 64
4.3. **Binary Potentials.** Source Images: LINEMOD ACCV dataset (Author: S. Hinterstoisser, Published: 2012, CC BY 4.0 License) . . . 66

4.4. **Illustrating Optimization Stage One.** 68
4.5. **Illustrating Optimization Stage Two.** 69
4.6. **Failure Case.** Source Images: LINEMOD ACCV dataset (Author: S. Hinterstoisser, Published: 2012, CC BY 4.0 License) 72
4.7. **Results - Global Hypothesis Generation** Source Images: LINEMOD ACCV dataset (Author: S. Hinterstoisser, Published: 2012, CC BY 4.0 License) 74

A.1. **Occlusion Extension of [8].** Source Images: LINEMOD ACCV dataset (Author: S. Hinterstoisser, Published: 2012, CC BY 4.0 License) 86
A.2. **Occlusion Extension (Our).** Source Images: LINEMOD ACCV dataset (Author: S. Hinterstoisser, Published: 2012, CC BY 4.0 License) 88
A.3. **Articulated Objects Dataset.** 89

BIBLIOGRAPHY

[1] A. Aristidou and J. Lasenby. Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5), 2011.

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

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110, 2008.

[4] J. Behley, V. Steinhage, and A. B. Cremers. Laser-based segment classification using a mixture of bag-of-words. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[5] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr. A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, 87(1), 2009.

[6] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992.

[7] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 10(6), 1988.

[8] E. Brachmann, A. Krull, F. Michel, J. Shotton, S. Gumhold, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[9] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother. Dsac - differentiable ransac for camera localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] L. Breiman. Random forests. *Machine Learning*, 45, 2001.

[12] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Pacific Grove, 1984.

[13] M. Brown and D. G. Lowe. Unsupervised 3d object recognition and reconstruction in unordered datasets. In *In Proceedings of the International Conference on 3-D Digital Imaging and Modeling (3DIM)*. IEEE Computer Society, 2005.

[14] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6), 1986.

[15] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, L. D. Stefano, and P. H. S. Torr. On-The-Fly Adaptation of Regression Forests for Online Camera Relocalisation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] L.-C. Chen, A. G. Schwing, A. L. Yuille, and R. Urtasun. Learning deep structured models. In *In International Conference on Machine Learning (ICML)*, 2015.

[17] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 18(3):265–298, 2004.

[18] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1. IEEE, 2005.

[19] P. Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 1996.

[20] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim. Recovering 6d object pose and predicting next-best-view in the crowd. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016.

[21] B. Drost, M. Ulrich, P. Bergmann, P. Härtinger, and C. Steger. Introducing mvtec itodd — a dataset for 3d object recognition in industry. In *In Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, 2017.

[22] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[23] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 1972.

[24] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361, 2017.

[25] S. P. Engelson and D. V. McDermott. Error correction in mobile robot map learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1992.

[26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

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

[28] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[29] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1), 1973.

[30] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[31] J. Gall, A. Yao, N. Razavi, L. J. V. Gool, and V. S. Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(11):2188–2202, 2011.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[33] I. Gordon and D. G. Lowe. *What and Where: 3D Object Recognition with Accurate Pose*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[34] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, 1988.

[35] S. Hinterstoisser, C. Cagniart, S. Ilic, P. F. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(5):876–888, 2012.

[36] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, editors, *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2011.

[37] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. R. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2012.

[38] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige. Going further with point pair features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[39] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In *In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[40] T. Hodaň, J. Matas, and Š. Obdržálek. On evaluation of 6d object pose estimation. In *In Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, 2016.

[41] T. Hodaň, X. Zabulis, M. Lourakis, Š. Obdržálek, and J. Matas. Detection and fine 3d pose estimation of texture-less objects in RGB-D images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

[42] D. Hoiem, C. Rother, and J. M. Winn. 3d layoutcrf for multi-view object class recognition and segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[43] B. K. P. Horn. Determining lightness from an image. *Computer graphics and image processing*, 3, 1974.

[44] D. P. Huttenlocher, G. A. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(9), 1993.

[45] International Civil Aviation Organization (ICAO) Montréal Quebec Canada. *Advanced Surface Movement Guidance and Control Systems (A-SMGCS) Manual. Doc 9830*, 2004.

[46] A. G. Ivakhnenko and V. G. Lapa. *Cybernetic Predicting Devices.* CCM Information Corporation., 1965.

[47] A. G. Ivakhnenko and V. G. Lapa. *Cybernetics and forecasting techniques.* American Elsevier, NY., 1967.

[48] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression tree fields - an efficient, non-parametric approach to image labeling problems. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2012.

[49] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[50] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5), 1976.

[51] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(1), 1960.

[52] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*, pages 1–30, 2015.

[53] N. Karlsson, E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM algorithm for robust localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[54] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[55] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[56] W. Kehl, F. Tombari, S. Ilic, and N. Navab. Real-time 3d model tracking in color and depth on a single cpu core. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[57] W. Kehl, F. Tombari, N. Navab, S. Ilic, and V. Lepetit. Hashmod: A hashing method for scalable 3d object detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[58] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2015.

[59] H. G. Kenngott, M. Wagner, M. Gondan, F. Nickel, M. Nolden, A. Fetzer, J. Weitz, L. Fischer, S. Speidel, H.-P. Meinzer, D. Böckler, M. W. Büchler, and B. P. Müller-Stich. Real-time image guidance in laparoscopic liver surgery: first clinical experience with a guidance system based on intraoperative ct imaging. *Surgical Endoscopy*, 28(3), 2014.

[60] A. Kirillov, D. Schlesinger, S. Zheng, B. Savchynskyy, P. H. S. Torr, and C. Rother. Joint training of generic cnn-crf models with stochastic optimization. In S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, editors, *Proceedings of the Asian Conference on Computer Vision (ACCV)*, volume 10112 of *Lecture Notes in Computer Science*, 2016.

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

[62] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts-a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(7), 2007.

[63] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Proceeding of the Conference Neural Information Processing Systems (NIPS)*, 2011.

[64] A. Krull, E. Brachmann, F. Michel, M. Y. Yang, S. Gumhold, and C. Rother. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[65] A. Krull, E. Brachmann, S. Nowozin, F. Michel, J. Shotton, and C. Rother. Poseagent: Budget-constrained 6d object pose estimation via reinforcement learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[66] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother. 6-dof model based tracking via object coordinate regression. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2014.

[67] J. J. Leonard and D. H. Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1991.

[68] V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[69] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.

[70] J. Liu, X. Gong, and J. Liu. Guided inpainting and filtering for kinect depth maps. In *International Conferhinterence on Pattern Recognition (ICPR)*. IEEE Computer Society, 2012.

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

[72] D. G. Lowe. Local feature view clustering for 3d object recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2001.

[73] D. Marr. *Vision. A computational investigation into the human representation and processing of visual information*. Freeman, San Francisco, 1982.

[74] M. Martinez, A. Collet, and S. S. Srinivasa. Moped: A scalable and low latency object recognition and pose estimation system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010.

[75] A. S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(10), October 2006.

[76] A. S. Mian, M. Bennamoun, and R. Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2), September 2010.

[77] F. Michel, E. Alexander Kirillov, Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, and C. Rother. Global hypothesis generation for 6d object pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[78] F. Michel, A. Krull, E. Brachmann, M. Y. Yang, S. Gumhold, and C. Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[79] J. Mund, F. Michel, F. Dieke-Meier, H. Fricke, L. Meyer, and C. Rother. Introducing lidar point cloud-based object classification for safer apron operations. In *In Proceedings of the International Symposium on Enhanced Solutions for Aircraft and Vehicle Surveillance Applications ESAVS*, 2016.

[80] R. M. Neal. Slice sampling. *Annals of Statistics*, 31, 2003.

[81] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[82] D. Nistér. Preemptive ransac for live structure and motion estimation. In *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2003.

[83] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4), 2011.

[84] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, editors, *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2011.

[85] C. F. Olson and D. P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1), 1997.

[86] C. Papazov and D. Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, volume 6492 of *Lecture Notes in Computer Science*. Springer, 2010.

[87] K. Pauwels, L. Rubio, and E. Ros. Real-time model-based articulated object pose detection and tracking with variable rigidity constraints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, 2014.

[88] S. Pellegrini, K. Schindler, and D. Nardi. A generalisation of the icp algorithm for articulated bodies. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2008.

[89] B. Pepik, P. Gehler, M. Stark, and B. Schiele. $3d^2pm$ – 3d deformable part models. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer Berlin Heidelberg, 2012.

[90] B. Pepik, M. Stark, P. V. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[91] C. J. Phillips, M. Lecce, and K. Daniilidis. Seeing glassware: from edge detection to pose estimation and shape recovery. In *Robotics: Science and Systems*, 2016.

[92] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[93] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[94] C. Rennie, R. Shome, K. E. Bekris, and A. Ferreira De Souza. A dataset for improved rgbd-based object detection and pose estimation for warehouse pick--and-place. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, Stockholm, Sweden, 02/2016 2016.

[95] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA, 2013.

[96] L. G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Instituted of Technology, 1963.

[97] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2), 2009.

[98] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009.

[99] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[100] T. Sattler, B. Leibe, and L. Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39(9), 2017.

[101] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1-3), 2002.

[102] P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1), 1966.

[103] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *Computing Research Repository (CoRR)*, 2015.

[104] T. Sharp, C. Keskin, D. P. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. W. Fitzgibbon, and S. Izadi. Accurate, robust, and flexible real-time hand tracking. In B. Begole, J. Kim, K. Inkpen, and W. Woo, editors, *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 3633–3642. ACM, 2015.

[105] A. Shekhovtsov. Maximum persistency in energy minimization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[106] A. Shekhovtsov, P. Swoboda, and B. Savchynskyy. Maximum persistency via iterative relaxed inference with graphical models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[107] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[108] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[109] I. Skrypnyk and D. G. Lowe. Scene modelling, recognition and tracking with invariant image features. In *In Proceedings of the International Symposium on*

Mixed and Augmented Reality (ISMAR), pages 110–119. IEEE Computer Society, 2004.

[110] S. Speidel, M. Delles, C. Gutt, and R. Dillmann. Tracking of instruments in minimally invasive surgery for surgical skill analysis. In G.-Z. Yang, T. Jiang, D. Shen, L. Gu, and J. Yang, editors, *Proceedings of the Third International Workshop on Medical Imaging and Augmented Reality*. Springer Berlin Heidelberg, 2006.

[111] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart. A layered approach to people detection in 3d range data. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2010.

[112] J. Sturm, A. Jain, C. Stachniss, C. C. Kemp, and W. Burgard. Operating articulated objects based on experience. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[113] P. Swoboda, C. Rother, H. Alhaija, D. Kainmüller, and B. Savchynskyy. A study of Lagrangean decompositions and dual ascent solvers for graph matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[114] P. Swoboda, B. Savchynskyy, J. H. Kappes, and C. Schnörr. Partial optimality by pruning for map-inference with general graphical models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[115] P. Swoboda, A. Shekhovtsov, J. Kappes, C. Schnörr, and B. Savchynskyy. Partial Optimality by Pruning for MAP-Inference with General Graphical Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2016.

[116] R. Szeliski. *Computer vision algorithms and applications*. Springer, London; New York, 2011.

[117] J. Taylor, J. Shotton, T. Sharp, and A. W. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[118] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim. Latent-class hough forests for 3d object detection and pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[119] Trimble Inc. SketchUp 3D Warehouse. https://3dwarehouse.sketchup.com/.

[120] C. von Hofsten and K. Lindhagen. Observations on the development of reaching for moving objects. *Journal of Experimental Child Psychology*, 28(1), 1979.

[121] G. Wahba. A least squares estimate of satellite attitude. *SIAM review*, 7(3), 1965.

[122] C. Wang and R. Zabih. Relaxation-based preprocessing techniques for markov random field inference. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[123] L.-C. T. Wang and C. C. Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4), 1991.

[124] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[125] J. M. Wong, V. Kee, T. Le, S. Wagner, G. L. Mariottini, A. Schneider, L. Hamilton, R. Chipalkatty, M. Hebert, D. M. S. Johnson, J. Wu, B. Zhou, and A. Torralba. Segicp: Integrated deep semantic segmentation and pose estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[126] C. Zach, A. Penate-Sanchez, and M.-T. Pham. A dynamic programming approach for fast and and robust object pose recognition from range images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[127] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[128] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2015.