Imperial College London

ACTIVE RECOGNITION AND DOMAIN ADAPTATION FOR 6D OBJECT POSE ESTIMATION

JUIL SOCK

Thesis submitted for the degree of Doctor of Philosophy

Supervised by DR. TAE-KYUN KIM

Imperial Computer Vision and Learning Lab
Department of Electrical and Electronic Engineering
Imperial College London

September 2020

ORIGINALITY DECLARATION

I hereby declare that this thesis and the work herein detailed, was composed and originated by myself, except where appropriately referenced and credited. Main ideas and some figures have been published in the following listed papers involving the author:

- **Juil Sock**, S.Hamidreza Kasaei, Luis Seabra Lopes, Tae-Kyun Kim, Multi-view 6D Object Pose Estimation and Camera Motion Planning using RGBD Images, Proc. of Internation Conference on Computer Vision Workshop (ICCVW), Venice, Italy, Oct 2017

- **Juil Sock**, Kwang-In Kim, Caner Sahin, Tae-Kyun Kim, Multi-Task Deep Networks for Depth-Based 6D Object Pose and Joint Registration in Crowd Scenarios, Proc. of British Machine Vision Conference (BMVC), Newcastle upon Tyne, UK, Sep 2018

- **Juil Sock**, Guillermo Garcia-Hernando, Tae-Kyun Kim, Active 6D Multi-Object Pose Estimation in Cluttered Scenarios with Deep Reinforcement Learning, Proc. of International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA, Oct 2020

- **Juil Sock**, Guillermo Garcia-Hernando, Anil Armagan, Tae-Kyun Kim, Introducing Pose Consistency and Warp-Alignment for Self-Supervised 6D Object Pose Estimation in Color Images, International Conference on 3D Vision(Under submission), Fukuoka, Japan, Nov 2020

COPYRIGHT DECLARATION

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution Non-Commercial No Derivatives license (CC BY-NC-ND).

Under this licence, you may copy and redistribute the material in any medium or format on the condition that; you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

ABSTRACT

Object pose estimation is an important task in computer vision with a wide range of applications including robotics and Augmented Reality. I investigate the problems of occlusion and lack of annotated real data in 6D object pose estimation in this thesis.

The main contributions of this thesis are building a 6D object pose estimation system for a crowded scene for both single image and multiple images, and a proposal of a new self-supervised learning framework.

The thesis first introduces a single-image pose estimation framework for a crowded scene, where the objects are under severe occlusions. Existing 6D pose estimators are unable to tackle such challenging scenarios, motivating the research towards end-to-end multi-task learning framework. A new training data generation framework is proposed which produces images of physically plausible object pose with a physics simulation.

The second contribution of this thesis explores how a strategic selection of camera movements can facilitate the task of 6D multi-object pose estimation for the crowded scenarios while respecting real-world constraints. The proposed agent learns the camera moves that produce the accurate object poses hypotheses for a given temporal and spatial budget.

The last contribution of this thesis is its introduction of self-supervised learning framework for domain adaptation for 6D object pose estimation. Conventionally, Deep Neural Network (DNN) based methods are trained primarily using rendered images of 3D object models. Since such methods do not generalise well to real testing data, DNNs also exploit real RGB images

with accurate 6D pose labels. The issue here is in a lack of real annotated data, since annotating 6D poses for 2D images in quality and quantity is hard. A novel self-supervised learning method is proposed, which results in a domain-invariant 6D pose estimator without pose labels of real data.

ACKNOWLEDGEMENTS

I would like to express my gratitude and appreciation for my supervisor Dr. Tae-Kyun Kim for the support and guidance. He introduced me to the field of computer vision and taught me how to research.

I am grateful to all of co-authors and collaborators with whom I worked with during my Ph.D., as well as all the members, past and present, of Imperial Computer Vision and Learning Lab for their companionship. The discussion and support I got were crucial to this thesis.

I thank all the friends I met during my Ph.D., particularly Daphne, Guillermo, Jong-Won, Jonny, Joseph, Pamela, Razvan, Sara, and Yang. You encouraged me and brought joy when I was going through a hard time. The journey would have been much tougher without you all.

I would like to thank all my family for their moral support and advice. I dedicate my thesis to my grandfather, who would have been very happy to see me graduate. Finally, I thank my loving wife Julia, for the endless support and encouragements, without which this adventure would never have been possible.

CONTENTS

1 INTRODUCTION 17
- 1.1 Challenges . . . 20
  - 1.1.1 Object-related challenge . . . 20
  - 1.1.2 Environment-related challenge . . . 22
- 1.2 Active vision . . . 23
- 1.3 Domain adaptation . . . 25
- 1.4 Thesis overview . . . 26
  - 1.4.1 Single-shot 6D object pose estimation . . . 27
  - 1.4.2 Long-term motion planning for object pose estimation . . . 28
  - 1.4.3 Self-supervised learning for object pose estimation . . . 29

2 RELATED WORK 31
- 2.1 Overview . . . 31
- 2.2 Object pose estimation . . . 32
  - 2.2.1 Pipelines . . . 32
  - 2.2.2 6D object pose problem formulation . . . 34
- 2.3 Active vision . . . 39
  - 2.3.1 Data acquisition . . . 40
  - 2.3.2 State update . . . 41
  - 2.3.3 View planning . . . 42
  - 2.3.4 Sensor motion planning . . . 44
- 2.4 Domain adaptation . . . 45

3 SINGLE-IMAGE 6D OBJECT POSE ESTIMATION 49
- 3.1 Overview . . . 49
- 3.2 Related work . . . 51

3.3 Proposed method 54
3.4 Experimental evaluation 64
3.5 Summary 68

4 LONG-TERM MOTION PLANNING FOR OBJECT POSE ESTIMATION 71
4.1 Overview 71
4.2 Related work 73
4.2.1 Multi-view object recognition 73
4.2.2 Active vision, object detection and poses. . 73
4.2.3 Attention to objects. 74
4.3 Proposed method 74
4.3.1 6D object pose estimation 76
4.3.2 Multi-view hypotheses fusion 77
4.3.3 Object attention module 79
4.3.4 Policy learning 81
4.4 Experimental evaluation 83
4.4.1 Dataset and experimental setup 83
4.4.2 Baseline policies 84
4.4.3 Evaluation metrics 85
4.4.4 Synthetic dataset 85
4.4.5 Real world dataset 88
4.4.6 Ablation experiments 88
4.5 Summary 91

5 SELF-SUPERVISED LEARNING FOR OBJECT POSE ESTIMATION 93
5.1 Overview 93
5.2 Related work 95
5.2.1 Datasets 95
5.2.2 Real-synthetic domain gap 96
5.2.3 Self-supervised pose estimation 97
5.3 Proposed method 98
5.3.1 Self-supervised 6D Object Pose Learning . 99

5.3.2 Warp-Align Fine-Tuning Stage 104
5.4 Experimental evaluation 106
5.4.1 Implementation Details 107
5.4.2 Results 108
5.5 Summary 115

6 CONCLUSIONS AND FUTURE WORK 117

BIBLIOGRAPHY 121

LIST OF FIGURES

Figure 1.1 Comparison between 2D object detection and 6D object pose estimation problem . 18
Figure 1.2 Sample models with object-related challenges 20
Figure 1.3 Visualisation of environment-related challenges 21
Figure 1.4 Active vision for object pose estimation . 24
Figure 2.1 Pipeline of active vision system 40
Figure 3.1 The proposed multi-task learning network architecture 53
Figure 3.2 Training data generation process 55
Figure 3.3 Training dataset generation output 56
Figure 3.4 Multiple hypotheses generation 58
Figure 3.5 Visualisation of joint registration module 60
Figure 3.6 Qualitative results of top 20 hypotheses for the *Coffee cup* scenario 62
Figure 3.7 Qualitative results of top 20 hypotheses for the *bunny* scenario 63
Figure 3.8 Comparison with other methods 65
Figure 4.1 The proposed framework 75
Figure 4.2 Multi-view hypothesis fusion 78
Figure 4.3 Baseline policies 84
Figure 4.4 Visualisation of attended objects during a trajectory 86
Figure 4.5 Evaluation on real dataset 87
Figure 4.6 Performances with different hyperparameters 89

Figure 4.7 Relationship between attention, verification score and action 90
Figure 5.1 Illustration of the proposed framework . . 94
Figure 5.2 Proposed self-supervised framework . . . 100
Figure 5.3 Proposed warp-align framework 103
Figure 5.4 Qualitative results on LINEMOD and LINEMOD OCCLUSION 111

LIST OF TABLES

Table 3.1 Quantitative results 66
Table 4.1 Quantitative results 85
Table 5.1 Ablation study 108
Table 5.2 Parameter Analysis 109
Table 5.3 Quantitative results on LINEMOD dataset with ADD metric 110
Table 5.4 Comparison with state-of-the-art 112
Table 5.5 Quantitative results on LINEMOD Occlusion 113
Table 5.6 Quantitative results on HomebrewedDB . 115

1

INTRODUCTION

Understanding the environment from images is a fundamental problem in machine perception and an important element for building an intelligent agent. For instance, pedestrian detection can help autonomous vehicle avoid accidents and image segmentation can remove background for video conference. This thesis focuses on the estimation of 6D pose of objects from images, which is an important problem in computer vision with a wide range of applications.

Before proceeding to a more detailed discussion, it is necessary to define 6D object pose problem. 6D pose consists of two parts: position and orientation. Object position is defined as the 3D displacement from the sensor, and it should be distinguished from 2D object detection problem (Russakovsky et al., 2015). As shown in Figure. 1.1, 2D object detection localises objects on the images space, whereas 3D displacement is in the 3D camera coordinate space. Unlike 2D detection problem which is agnostic to the sensor used to produce the image, estimating 3D position from 2D image requires additional camera model parameters such as intrinsic or distortion parameters to relate the image space to sensor coordinate space. Orientation can be parameterised in various formats such as Euler angle, Quaternion or rotation matrix. Euler angle has the Gimbal lock problem which occurs when the two axes are positioned in a parallel configuration,



((a)) ((b))

Figure 1.1: Comparison between 2D object detection and 6D object pose estimation problem. (a) 2D object detection problem localises objects on the image plane. In this example, the task is to detect the drill on the table and the output is represented as the pixel position of a box enclosing the object highlighted in red. (b) 6D object pose estimation localises the object in 3D space as well as estimate orientation. The camera is visualised with the axis to indicate the viewpoint. The estimation is represented as 3D bounding box enclosing the object highlighted in red.

resulting infinite solution for the same pose. Quaternions have four parameters with a scalar component and three dimensional vector component. Quaternions do not have the Gimbal lock problem but still have an ambiguity as **q** and **-q** represent the same rotation. Directly predicting rotation matrix has the risk of producing invalid rotation. Finding the best representation for 3D orientation is an active area of research with recent works proposing 5D or 6D continuous representation (Zhou et al., 2019). The minimum number of parameters to represent a 3D angle is three, therefore, at least 6 parameters (3 for displacement and 3 for orientation) are required to describe the rigid transformation. Hence it is called 6D pose estimation. The reference pose can be defined in either the sensor coordinates or world coordinates depending on the applications, although sensor coordinates are widely used in most of the literature (Brachmann et al., 2016; Hinterstoisser et al., 2011; Park et al., 2019; Rad and Lepetit,

2017; Tejani et al., 2014) and dataset (Doumanoglou et al., 2016; Hinterstoisser et al., 2011; Hodaň et al., 2017; Tejani et al., 2014).

6D object pose estimation finds its applications in many different areas. Robotics manipulation can benefit from accurate 6D object pose information. In particular, picking and grasping objects with a robotic manipulator (Holz et al., 2014; Kumra and Kanan, 2017; Zhu et al., 2014) can plan the arm and gripper manipulation based on the estimated pose of the object. Recently, Amazon Picking Challenge demonstrated how object pose estimation could be used in a realistic warehouse environment. Augmented Reality (AR) (Lindlbauer and Wilson, 2018) technology estimates the object pose to add computer-generated contents. By localising and estimating the pose of an object, more realistic augmentation can be generated, *i.e.* changing colour or texture of the object in the image. 6D pose estimation in augmented reality application has potentially large impact economically. For instance, it can be applied in a construction site where 6D pose of the building can be estimated from partially built structure, which can be used to instruct builders for any mistake or provide guidance. This can increase the productivity and efficiency, which can reduce the cost. Object pose estimation is also beneficial for autonomous driving technology to monitor the nearby vehicles to avoid the collision.

Object pose estimation is used in different context depending on the applications, as each application requires a different level of accuracy and degree of freedom. Pose estimation in autonomous driving is more tolerable to the orientation estimation errors compared to AR applications or robotic manipulations. Sometimes full 6D pose is not required, and only yaw angle is estimated as one can assume the vehicles on the road are always in up-right position (Liu et al., 2019).

The focus of this thesis is the accurate estimation of full 6D object pose *given* the rigid models of the object from 2D image. The

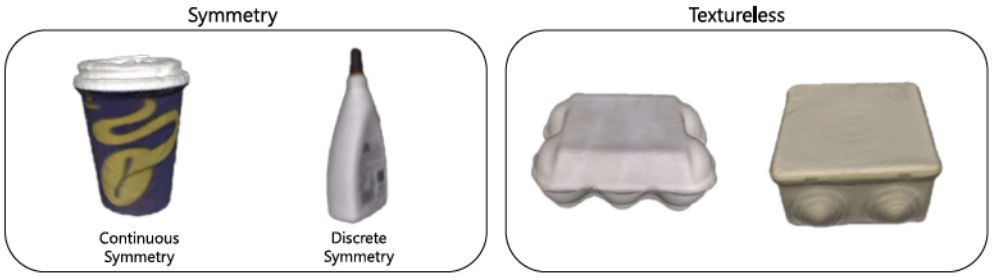


Figure 1.2: Sample models with object-related challenges. (Left) Symmetric objects do not have one-to-one mapping between pose parameters and input image which makes pose estimation more difficult. (Right) Colour is a strong discriminative feature for image-based object pose estimator. Textureless objects do not have colour pattern and only provides edge features.

problem is referred as instance-level 6D object pose estimation, opposed to category-level pose estimation where the goal is to estimate the pose of any object belonging to the same class rather than a given instance (Sahin and Kim, 2018). With the object model, there exist a unique 6D pose parameters to be estimated for a given image except when the model is symmetric.

1.1 CHALLENGES

Any method for 6D object pose estimation has to overcome challenges to robustly work in diverse environments. The main challenges can be categorised into two, depending on the source: object-related and environment-related.

1.1.1 *Object-related challenge*

object-related challenges are a set of problems related to the object of which the pose is to be estimated. Figure. 1.2 provides example objects with such challenge.

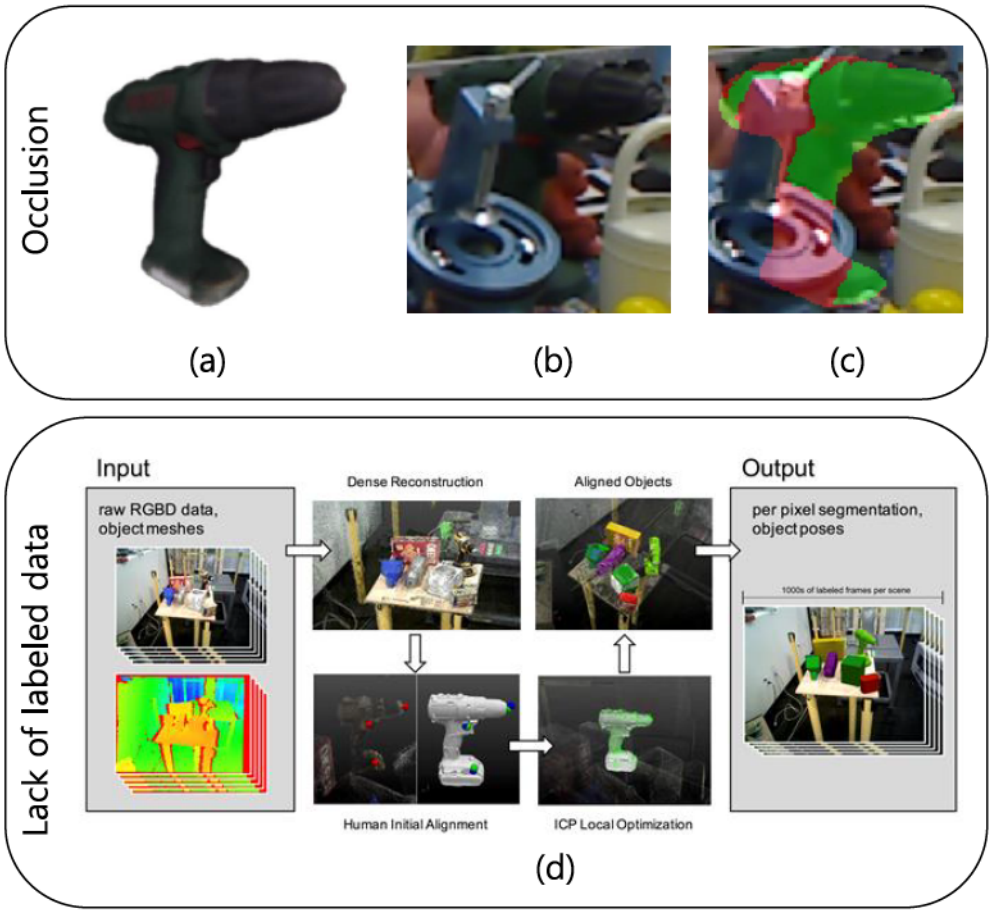


Figure 1.3: Visualisation of environment-related challenges. (a) The object of interest. (b) object with occlusion and background clutter. Useful features for visual recognition, such as brand logos are invisible. (c) Visible and occluded regions are highlighted green and red, respectively. (d) Generation of a large quantity of high quality pose label annotation for images is difficult. Labelfusion (Marion et al., 2018) proposes a pipeline to annotate multi-view images with a sequence of stages such as dense reconstruction, manual alignment and ICP alignment. However, such a strategy is restricted to a static environment, and error can accumulate from each stage.

SYMMETRY Many objects, such as industrial parts, exhibit symmetries. Symmetry causes multiple, even infinite, pose transformations to be identical. For instance, a cylinder is infinitely symmetric around the axis of symmetry and a hexagon head bolt has discrete rotational symmetry. This results in non one to one mapping for symmetric object and needs to be tackled adequately for estimation to be robust. Kehl et al. (2017) have avoided the problem by formulating the rotation estimation as a classification problem. Possible rotation is divided into a discrete class where the classes around the axis of symmetry are merged to the same class. Sundermeyer et al. (2018) have avoided inferring pose parameter directly by matching input image embedding to template embedding. Park et al. (2019) have proposed transformer loss to handle symmetry by guiding predictions to the closest symmetric pose.

LACK OF VISUAL FEATURES Texture is a useful feature for pose estimation, especially for methods which only uses RGB images. Texture provides a discriminative feature which can help pose estimator to distinguish different poses. Hence pose estimator designed for textureless objects strongly rely on other modality such as depth map which captures geometric information (Hodaň et al., 2017). Edge and surface normal are popular features for textureless object pose estimation (Hodaň et al., 2015; Tombari et al., 2013; Zhang and Cao, 2017).

1.1.2 *Environment-related challenge*

Environment-related challenges are a set of problems related to the environment where pose estimation is taking place. These challenges are the main focus of this thesis. Figure. 1.3 visualises the challenges.

OCCLUSION Occlusion occurs when the view of the object is partly or entirely blocked by another object in the scene. Occlusion occurs in a realistic environment and can degrade the pose estimation performance. It is a common scenario for pick and place operation in the industry where there are many identical parts piled in a box. Occlusion can be handled by explicitly detecting the foreground object (Park et al., 2019; Peng et al., 2019) or engineering a part-based approach along with voting (Doumanoglou et al., 2016; Tejani et al., 2014). The problem is more significant when the foreground object and the occluding object are different instances of the same object.

LACK OF REAL ANNOTATED DATASETS In recent years, DNN have demonstrated to be a powerful tool to learn the mapping from the input image to pose parameters. DNN is a highly data-driven machine learning algorithm, and a large amount of training data is required to learn features robust to variability in an image. However, acquiring adequate annotation in quality and quantity is very time-consuming and labour-intensive. Furthermore, the pose estimation accuracy degrades if the environment where the training dataset is captured is substantially different from the test dataset (Kaskman et al., 2019). Consequently, the lack of real annotated data causes the learned pose estimator to overfit to the small amount of annotated real data.

1.2 ACTIVE VISION

In this thesis, Chapter 3 and Chapter 4 tackle the occlusion challenge introduced in the previous section. A bin-picking environment where multiple objects are stacked in a confined space are used in the chapters. The bin-picking scenario provides various degree of occlusion and distractions for the estimator, therefore suitable to highlight the challenge. Chapter 3 proposes a frame-

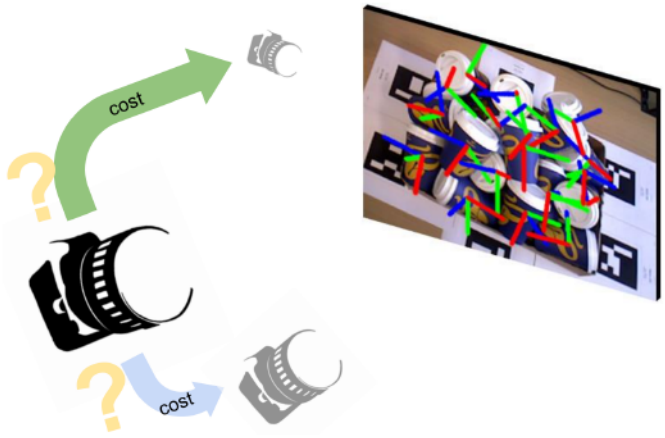


Figure 1.4: Active vision for object pose estimation. If the camera can move to a different view, the new perspective can give extra information about the objects which were previously unavailable because of the occlusion. In this case, the agent can choose to move the camera to either a green position or blue position to detect more objects. Upon decision, cost is another factor that should be considered.

work which estimates poses of multiple objects from a single image. However, the quantity and quality of the information provided in one image may not be enough to achieve satisfactory accuracy in the presence of severe occlusion. The amount of information contained in a single image is limited by the sensor range and FoV (Field-of-view) of the sensor. The quality of information may not be enough to confidently predict the object pose if the captured portion of the object lacks strong feature. This motivates the integration of 6D pose estimation problem with active vision framework in Chapter 4.

Active vision refers to the capability to actively adjust visual sensor position to obtain useful information for the given task. It enables the system to analyse current data and sequentially select views (view-planning) to maximise the quality of the estimation. In this thesis, an active vision system for 6D pose estimation via reinforcement learning framework is proposed in Chapter 4. The

application of active vision to multiple 6D object pose estimation problem is visualised in Figure. 1.4. The chapter explores how moving the camera in the scene, reaching different viewpoints, can help to overcome the occlusion challenges in 6D object pose estimation, identifying two ends of the spectrum of such a problem. On one end, one can decide not moving the camera at all, reducing the problem to single-shot object pose estimation as presented in Chapter 3. On the other end, the camera can be moved to cover all the possible viewpoints (Zeng et al., 2017). However, this does not respect real-world constraints, such as time limitation in terms of a number of movements and energy expenditure in terms of distance travelled. Chapter 4 is about finding a compromised solution that moves the camera strategically by reaching the most informative viewpoints given a limited number of camera movement and keeping the travelled distance as low as possible.

1.3 DOMAIN ADAPTATION

The other environment-related challenge in Section 1.1 is the lack of real annotated datasets. To overcome the challenge, there are two major approaches that can reduce the cost of collecting and annotating more data: semi-supervised learning and transfer learning. Semi-supervised learning exploits the unannotated data together with annotated data and transfer learning exploits the similarity in the data between domains and transfer knowledge (Csurka, 2017). Semi-supervised learning requires a set of annotated images it assumes the annotated and unannotated data are drawn from the same distribution. In contrast, transfer learning does not have the same assumption. Instead, it attempts to learn a classifier from the source domain (training data with annotation) and transfer to the target domain (unseen or unlabelled test data). Domain adaptation (DA) is a particular case

of transfer learning which assumes that the task is the same, *i.e.* same class labels for both domains. For instance, face recognition trained with passport photos with a clean background (source domain) may not perform well when tested with a face image cropped from CCTV footage with noises and background clutter (target domain). Degradation in performance is due to the difference in the distribution of the two domains, called domain shift or domain gap.

Chapter 5 uses training datasets from two domains: source domain corresponding to synthetic rendered images with automatically generated annotation; and target domain corresponding to real images with no pose annotation. Instead of fully relying on domain randomization technique, a self-supervised learning framework is proposed to reduce the domain gap and learn domain-invariant 6D object pose estimator.

1.4 THESIS OVERVIEW

The thesis addresses the environment-related challenges of 6D object pose estimation. Occlusion problem is tackled in two different scenarios: single-shot 6D object pose estimation where poses are estimated from a single image; and 6D pose estimation in an active setting where multiple views of the scene are available and the selection of viewpoint is allowed. Lack of real annotated dataset problem is investigated, and a novel self-supervised learning framework which can learn pose estimation without pose label is proposed. Before discussing the proposed methods, chapter 2 introduces the general literature review on related areas. Chapter 3 to 5 contain the main contributions, which are summarised below. Chapter 6 concludes the thesis with discussion and future work.

Here is the summary and contributions of the chapters:

1.4.1 *Single-shot 6D object pose estimation*

In bin-picking scenarios, multiple instances of the same object are stacked in a pile *randomly*. Hence, the instances are inherently subjected to the challenges: *severe occlusion*, *clutter*, and *similar-looking distractors*. Most existing methods are, however, for single isolated object instances, while some methods tackle crowd scenarios as post-refinement, which accounts for multiple object relations. In this chapter, I address recovering 6D poses of multiple instances in bin-picking scenarios in depth modality via multi-task learning in DNNs. The proposed architecture *jointly* learns multiple sub-tasks: 2D detection, depth, and 3D pose estimation of individual objects; and joint registration of multiple objects. For training data generation, depth images of physically plausible object pose configurations are generated by a 3D object model in a physics simulation, which yields diverse occlusion patterns to learn. I adopt a state-of-the-art object detector to generate bounding boxes, and 2D offsets are further estimated via a network to refine misalignments in 2D detections. Additionally, the depth and 3D pose estimators are designed to generate multiple hypotheses per detection. This allows the joint registration network to learn occlusion patterns and remove physically implausible pose hypotheses. I apply the proposed architecture on both synthetic (own generated dataset and Siléane dataset) and real (a public Bin-Picking dataset) data, showing that it outperforms state-of-the-art methods. In summary, the main contributions are :

- A training dataset generation framework which can generate realistic occlusion pattern for bin-picking scenario is proposed.
- A single pose estimation network that can perform object detection, pose estimation, and joint registration simultane-

ously is proposed. The algorithm is tested and compared with different objects and baselines.

1.4.2 *Long-term motion planning for object pose estimation*

In this chapter, I explore how a strategic selection of camera movements can facilitate the task of 6D multi-object pose estimation in cluttered scenarios while respecting real-world constraints important in robotics and augmented reality applications, such as time and distance travelled. In the proposed framework, a set of multiple object hypotheses generated is given to an agent, which is inferred by an object pose estimator and subsequently spatio-temporally selected by a fusion function that makes use of a verification score that circumvents the need of ground-truth annotations. The agent reasons about these hypotheses, directing its attention to the object which it is most uncertain about, moving the camera towards such an object. Unlike previous works that propose short-sighted policies, the proposed agent is trained in simulated scenarios using reinforcement learning, resulting in a non-myopic policy. It attempts to learn the camera moves that produce the most accurate object poses hypotheses for a given temporal and spatial budget, without the need of viewpoints rendering during inference. The proposed method assumes the objects in the scene are static. The experiments show that the proposed approach successfully estimates the 6D object pose of a stack of objects, showing superior performance compared to the baselines. In summary, the main contributions of this chapter are:

- An active 6D multi-object pose estimation within a reinforcement learning framework is proposed. State and action spaces are carefully designed, and a reward function is tailored to the problem of interest.

- Application-specific modules are proposed: a hypotheses fusion function that works without ground-truth annotations by a proposed verification score and an object-specific attention module that directs the attention of the agent.
- The proposed method and baselines are evaluated on synthetic and real cluttered scenarios.

1.4.3 *Self-supervised learning for object pose estimation*

Recent approaches for 6D object pose estimation typically train a DNN on real training images which are generally expensive to annotate. A common workaround is to resort to synthetic data, with the drawback of poor generalization to real images mainly because of a domain gap. In this chapter, I introduce a self-supervised approach for training a 6D object pose estimator. The basic idea of self-supervised learning is to automatically generate supervisory signal to solve the task without manually annotated data. The proposed approach can learn without labelled real images or depth information with the aid of a pose consistency loss. This loss is computed using a differentiable renderer that generates synthetic images based on a preliminary pose estimation on real images. Minimizing the discrepancy between the estimation of synthetic and real images guides the learning process towards better generalization. Additionally, a warp-align stage is proposed that fine-tunes the previously trained model by enforcing the photometric consistency between pairs of different object views, where one image is warped and aligned to match the view of the other and thus enabling their comparison. The proposed approach is flexible such that various off-the-shelf 6D object pose estimators can be used as a baseline model for the self-supervision, and I report a boost in performance on the two baselines that I tried. The proposed method exhibits state-of-the-

art performance when compared to recent methods that are on synthetic data, domain adaptation baselines and a concurrent self-supervised approach in the absence of real pose annotations and depth data. I report results on three popular benchmarks: LINEMOD, LINEMOD OCCLUSION, and HomebrewedDB. The contributions of this chapter can be summarised as follows:

- A self-supervised framework is proposed that is flexible to use any off-the-shelf 6D object pose estimator as a baseline model by leveraging pose consistency in the synthetic and real domains.
- The proposed approach only makes use of monocular RGB images both in training and testing of the estimator as opposed to previous self-supervised works.
- The effectiveness of making use of the camera pose estimations to fine-tune the initial pose estimator is shown.
- The experiments investigate each step of the proposed framework in detail and show their effectiveness on LINEMOD (Hinterstoisser et al., 2012), LINEMOD OCCLUSION (Hinterstoisser et al., 2012) and HomebrewedDB (Kaskman et al., 2019) datasets.

2

RELATED WORK

2.1 OVERVIEW

The thesis covers various scenarios for 6D object pose estimation each requiring different approaches. Multiple areas of research are involved in the thesis, and hence this chapter is divided into three sections. Please note that this chapter provides a general literature review on each research area. Each chapter has its related work section, which discusses existing works that are more related to the proposed method in the chapter.

The chapter begins with a review of methods for 6D object pose estimation from a single image. The second section reviews active vision literature, as the systems in Chapter 4 involves selecting camera viewpoints for a bin-picking scenario. Chapter 5 tackles the issue with the lack of annotated real data for 6D object pose estimation by proposing a domain adaptation technique. Hence a review on domain adaptation for DNN is provided in the last section.

I refer to the following survey papers for a more exhaustive list: Sahin et al. (2020), Zeng et al. (2020) and Wang, Devin, Cai, Yu and Darrell (2019).

2.2 OBJECT POSE ESTIMATION

This section reviews relevant works on object pose estimation from a single RGB/-D images. The first part discusses the necessary components of 6D pose estimation and introduces different pipelines to integrate those components. The second part groups existing methods into categories based on how the 6D pose problem is formulated.

2.2.1 *Pipelines*

Typically, 6D object pose estimation from RGB/-D consists of three major components: object detection, pose estimation and refinement. Detection determines the location of the object in the 2D image plane, usually represented by the bounding box enclosing the visible part of the object. Pose estimation determines the rotation and translation of the object in 3D space, and the refinement module refines the hypothesis to better align with the data. This section summarises how different methods solve those components separately or jointly. Voting-based methods is more robust to partial occlusion and noise compared to holistic methods which are often used with detection-based methods. However, it is more complex with more hyper-parameters to tune as it consists of more series of operations. Joint methods is often faster in execution as it learns and infers multiple tasks simultaneously.

VOTING-BASED METHODS: Voting-based methods (Birdal and Ilic, 2015; Doumanoglou et al., 2016; Drost et al., 2010; Tejani et al., 2014) consist of multiple stages: correspondence matching, pose clustering, and hypothesis selection. The process begins with finding correspondences between the reference points in the scene and the model points. Doumanoglou et al. (2016) and

Tejani et al. (2014) have used Random Forest to extract features and match the model templates to reference point in the scene. Birdal and Ilic (2015) and Drost et al. (2010) have used PPF (Point Pair Feature) and hash table for matching. Each correspondence casts a vote on 6D pose space (Hough space) where votes are aggregated from all the reference points. A clustering algorithm is used to group votes into pose clusters where they are ranked based on various criteria such as the cluster size or variance within each cluster. Each cluster produces a pose hypothesis by either averaging candidate poses in the cluster or selecting the most likely pose hypothesis.

DETECTION-BASED METHODS: Detection-based method utilises off-the-shelf 2D detectors, *e.g.* Faster R-CNN (Ren et al., 2015), SSD (Liu et al., 2016) or YOLO (Redmon and Farhadi, 2017), to first detect the object on an image and then perform separate pose estimation on the detected segment of the image (Balntas et al., 2017; Rad and Lepetit, 2017; Sundermeyer et al., 2018). The disadvantage of such approach is the slow execution time as the total run-time is the sum of the two separate stages.

JOINT METHODS: Joint methods (Kehl et al., 2017; Li et al., 2019; Peng et al., 2019; Tekin et al., 2018; Zakharov, Shugurov and Ilic, 2019) have gained more popularity in recent years, which follows the multi-task learning framework where multiple learning tasks are solved simultaneously, exploiting the commonalities between tasks. The methods in this group commonly have a shared low-level feature extractor where separate branches are attached for different tasks. The methods typically build on top of the popular 2D detectors. SSD6D (Kehl et al., 2017) has modified SSD (Liu et al., 2016) architecture and Wang, Sridhar, Huang, Valentin, Song and Guibas (2019) have added an extra branch

to Mask RCNN (He et al., 2017) to estimate Normalised Object Coordinate Space (NOCS) map.

REFINEMENT: Refinement is an optional component that is independently applied at the end of pose estimation. Given an initial hypothesis from the pose estimator, refinement module transforms the hypothesis into the input data space (either a monocular image or depth image in this case) and adjust the hypothesis, so it better aligns with the data. When the depth map is available, ICP (Iterative Closest Point) (Arun et al., 1987) is the most widely used. ICP iteratively applies transformations to the initial hypothesis until the sum of squared differences between two sets of point cloud is minimised. SSD6D (Kehl et al., 2017) has proposed an edge-based energy equation for refinement on RGB images. Recently, learning-based refinement approaches have shown to perform well. Densefusion (Wang, Xu, Zhu, Martín-Martín, Lu, Fei-Fei and Savarese, 2019) has proposed an iterative pose refinement network using both colour and depth image by estimating pose residual given the hypothesis and the input data. DPOD (Zakharov, Shugurov and Ilic, 2019), DeepIM (Li et al., 2018), and Manhardt et al. (2018) have proposed an RGB based refinement network to estimate residual pose given the input image and pose hypothesis.

2.2.2 *6D object pose problem formulation*

Pose estimation problem can be formulated in various ways, and this section introduces and discusses different approaches.

2.2.2.1 *Template matching*

Template matching methods match parts of an image with template images where each template has pose annotation. Matching is performed in feature space where templates and input data are represented as either hand-crafted features (Hodaň et al., 2015; Munoz et al., 2016; Payet and Todorovic, 2011; Tjaden et al., 2017; Ulrich et al., 2011) or learned from data (Balntas et al., 2017; Sundermeyer et al., 2018; Wohlhart and Lepetit, 2015; Zakharov et al., 2017).

HAND-CRAFTED FEATURE REPRESENTATION METHODS Traditional object pose estimation system utilises appearance features (*e.g.* contour, edges, colour gradient) and geometric features (surface normals, depth edge) to match the input data and templates. LINEMOD (Hinterstoisser et al., 2011) and Hodaň et al. (2015) have used surface normals and colour gradient to encode the data. At test time, sliding windows technique is used to compare every patch in the input image to every template. Such an approach inevitably scales poorly to the increasing number of templates and image size. Konishi et al. (2016) have utilised histogram of pixel gradient and proposed a hierarchical pose tree to accelerate the inference time. Tjaden et al. (2017) have proposed temporally consistent local colour histogram, which utilises temporal information to reduce the search space achieving real-time pose tracking. LINEMOD uses holistic template matching approach which uses the template with the full view of the object. Since the template only contains the foreground object, holistic template matching approaches suffer from occlusion and clutter.

LEARNING-BASED FEATURE REPRESENTATION METHODS Learning-based template matching methods (Balntas et al., 2017;

Sundermeyer et al., 2018; Wohlhart and Lepetit, 2015) have shown promising results in recent years. Wohlhart and Lepetit (2015) have proposed holistic 3D orientation estimation methods which learn discriminative descriptor by using triplet loss to train CNN (Convolutional Neural Network). Triplet loss maximises the Euclidean distance between intra-class sample in descriptor space and simultaneously minimises the distance between intra-class samples. However, the descriptor learned with such loss may not produce continuous manifold in the latent space. Balntas et al. (2017) have extended the work of Wohlhart and Lepetit (2015) by constraining the distance between the samples to be proportional to the difference in the orientation angle of the samples. Template matching generally suffers when distortion or variability is present in the test data, as generated templates are often clean without occlusion or perturbation. Sundermeyer et al. (2018) have proposed unsupervised 3D orientation estimation methods with an augmented autoencoder. The core idea is to use denoising autoencoders to reconstruct the clean image from the input image with variabilities such as lighting, occlusion and even small translation by considering them as noise. The resulting descriptor is matched with existing template descriptor in the latent space via nearest neighbour search.

2.2.2.2 *Classification*

3D orientation estimation methods can be formulated as classification. All 3D rotation which falls into a given small interval (bins) are considered to be the same class. For DNN architecture, orientation classification has been reported to be more stable during training than directly regressing roll, pitch and yaw parameters (Papon and Schoeler, 2015). Also, it has the benefit of producing multiple hypotheses (Gupta et al., 2015; Kehl et al., 2017; Poirson et al., 2016). SSD6D (Kehl et al., 2017) has used SSD (Liu et al., 2016) network to detect the object and used classi-

fication network to estimate the orientation of the detected object. Su et al. (2015) have trained classifiers to predictpose class for each of azimuth, elevation and in-plane rotation. However, the accuracy of the pose estimation with classification is fundamentally limited by the granularity of the bin due to quantisation error. Also the number of bins increases exponentially when the full 3D rotation is considered. Therefore using classification is more popular for partial orientation estimation with less degree of freedom. Papon and Schoeler (2015) have estimated the orientation of furniture around the floor's normal axis, reducing the problem to one dimension. Gupta et al. (2015) have proposed to classify only the azimuth of the object in the geocentric coordinate frame.

2.2.2.3 *Regression*

A growing portion of recent works on object pose estimation falls into this category. This category can be sub-categorised, depending on what is being regressed. The model can either directly regress the pose parameters or regress intermediate representation that can be efficiently transformed to the pose parameters.

DIRECT POSE PARAMETER REGRESSION PoseNet (Kendall et al., 2015) has directly regressed 3D position and 3D orientation of the camera given a monocular image of the scene. The pose is defined in an arbitrary global reference frame, and 3D orientation is parameterised as Quaternion. This work is one of the first to show that position and orientation regression can be learned on a single CNN, with the scale factor carefully chosen for optimal performance. Mahendran et al. (2017) have also directly regressed Quaternion for 3D pose estimation problem.

REGRESSION ON INTERMEDIATE REPRESENTATION Performance of direct pose parameter regression may be sub-optimal

since directly estimating 6D pose parameter has to solve complex out-of-image-plane regression problem. Several methods proposed to regress intermediate representation instead that is on the image plane, which transforms the problem into a well-studied Perspective-n-Point (PnP) problem. PnP estimates 6D pose of the calibrated camera given a set of 3D points in the world and the corresponding 2D projection in the image. There are two types of intermediate representation: the representation describing 2D projection given fixed 3D points (3D to 2D) and 3D points for a given pixel location (2D to 3D).

For 3D to 2D, 3D locations of control points are predetermined for each object model. The most popular choice for the control points is 3D bounding box enclosing the whole object model in its canonical pose (Li, 2017; Rad and Lepetit, 2017; Tekin et al., 2018). BB8 (Rad and Lepetit, 2017) has directly estimated pixel location of the projected eight 3D bounding box vertices. In PVNet (Peng et al., 2019), the authors have argued that the projected bounding box corners are, in most cases, far away from the pixel belonging to the object and this results in larger localisation error. Therefore the authors selected the keypoints on the object surface with the farthest point sampling algorithm and trained CNN that regresses unit vectors pointing to the keypoints for every pixel. Alternatively, Pavlakos et al. (2017); Tulsiani and Malik (2015); Zhou et al. (2018) and Oberweger et al. (2018) have used heatmap to estimate the keypoint location on the image plane.

For 2D to 3D, methods in this category estimates 3D vertex location of the object model for each pixel on the image, called object coordinate. In this line of work, inevitably two problems need to be solved simultaneously: foreground pixel classification and object coordinate estimation. This is necessary as only the pixels belonging to the object should contribute to estimating the

pose. Brachmann et al. (2014, 2016) have used Random Forest for object coordinate regression and Wang, Sridhar, Huang, Valentin, Song and Guibas (2019); Zakharov, Shugurov and Ilic (2019) and Li et al. (2019) have demonstrated that very accurate pose estimation can be achieved with or without depth map using CNN based model.

This section categorised and summarised different approaches to 6D object pose estimation. Hand-crafted features such as point-pair feature and LINEMOD features with color image and geometric data from depth image dominated the early methods. However, such methods suffered in the presence of occlusion and variation in data such as illumination change. More recently, neural network based data driven learning methods have gained popularity due to its robustness and flexibility to learn multiple tasks simultaneously. However, this new powerful model has introduced a new challenge as it requires large amount of annotated data which is expensive to create.

2.3 ACTIVE VISION

This section presents a general literature review on the active vision system. The general pipeline for an active vision system is discussed before describing recent development for different applications.

PIPELINE FOR ACTIVE SYSTEM Figure.2.1 shows the general pipeline for the active vision system. The pipeline includes data acquisition, state update, view planning and sensor motion planning. The process starts from a random sensor position, and each component is executed sequentially. The entire process is iterated until some termination criteria are satisfied. The termination criteria depend on the application; *e.g.* scan coverage for object reconstruction or class uncertainty for object recognition. The

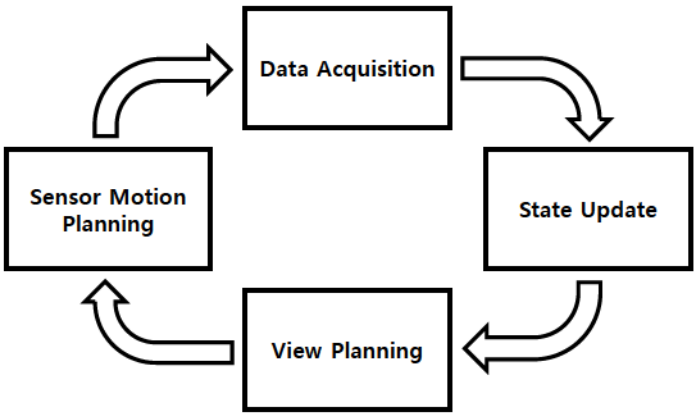


Figure 2.1: Pipeline of active vision system.

following sections describe each stage of the pipeline and discuss how existing approaches employ different strategies.

2.3.1 *Data acquisition*

Various sensors provide different type of information about the environment for decision making. Depth camera or RGB-D camera are often used for various tasks as it can provide appearance as well as geometric information. Geometric information is particularly useful in active vision, as many system involves manoeuvring the sensor in the 3D space (Bircher et al., 2016; Johns et al., 2016; Krainin et al., 2011; Kriegel et al., 2012; Liu et al., 2018). Pan-Tilt-Zoom camera is also used to take additional images of the detected object with higher resolution for verification (Coates and Ng, 2010). Monocular image is used more often for active object recognition task, as texture and colour could provide sufficient information about the object for recognition. (Browatzki et al., 2012; Grtner et al., 2020; Jayaraman and Grauman, 2016; Kiciroglu et al., 2020).

2.3.2 *State update*

Active vision system plans future viewpoints to acquire new data, based on both current and past information. Therefore a state representation is needed to accumulate the collected data and information.

Voxel is a popular representation for 3D data. Voxel tessellates 3D space into small cubes analogous to pixel in an image, which tessellates 2D image plane into square blocks. Each voxel can store information such as colour, surface normal and the number of 3D points. The main advantage of using voxel lies in its simplicity and it has been used for diverse tasks in active vision system from reconstruction (Cui et al., 2019; Delmerico et al., 2018; Dong et al., 2019; Mendoza et al., 2019) to recognition (Kriegel et al., 2013; Wu et al., 2015). However, voxel is not suitable for modelling complex scene or large space as memory usage becomes prohibitively expensive.

Point cloud stores spatial as well as geometric and colour information. Point cloud can preserve more fine details about the surface, however, it cannot represent information about unknown or empty spaces. It can be used to accumulate sensor data and converted to different representation during view planning. Han et al. (2019) have projected point cloud data to depth map which is used as an input to Deep Q-Network to infer the next best view.

Learned representation for 3D structure has drawn lots of attention in recent years. Instead of manually defining it, representation is learned from data for both the task and view planning. Cheng, Wang and Fragkiadaki (2018) have proposed geometry-aware network which integrates information across multiple views into 3D latent feature tensor via a recurrent network. The

latent feature aggregates sequence of RGB input images and is used for reconstruction, segmentation, classification of the object as well as camera policy.

2.3.3 *View planning*

View planning lies at the core of the active vision system. The common strategy for view planning is to sample a set of multiple candidate viewpoints and evaluate each candidate to form the optimum trajectory. The definition of optimum trajectory depends on the requirement and the task goal. But generally a view planning algorithm targets to find a sequence of views which can generate acceptable quality for the task with the minimum cost. The views are selected based on the current state as well as utility function which represents the expected information gain and the cost. The recent developments on the view planning algorithm for different applications are discussed in this section.

3D RECONSTRUCTION The objective of 3D reconstruction is to capture the shape and appearance of the real object or scene and reconstruct the realistic 3D model. This requires multiple views from different angles due to limited sensor range and occlusion. The overlap between the FoV of captured views should be carefully controlled: the region with the multiple overlaps can be reconstructed with higher fidelity by integrating sensory data from multiple sources but too much overlap can be wasteful. Early seminar works (Maver et al., 1993) on this topic involves determining the 3D points of object silhouettes under orthographic projection.

Isler et al. (2016) have compared and combined several metrics for information gain in the utility function, which includes visibility likelihood and the likelihood of seeing new parts of the

object. The work uses voxel representation where each voxel has estimated occupancy likelihood. The information gain is formulated with Shannon's entropy formula from which five different metrics are derived. Cui et al. (2019) have proposed an active 3D object reconstruction framework with multiple sensors and modality by building voxelised occupancy grid. Devrim Kaba et al. (2017) have proposed a reinforcement learning solution for view planning to maximise the view coverage on a given 3D model with the minimum number of views. By introducing a novel score function, the method demonstrates superior performance over greedy policy in almost all experiments. Dong et al. (2019) have proposed a multi-robot framework to perform collaborative scanning for scene reconstruction. Given the partially scanned scene, candidate views are sampled based on uncertainty map and validness map built during the operation and multiple robots are dynamically assigned the views to visit.

OBJECT RECOGNITION Object recognition identifies objects from the sensor data and it includes but not limited to detection, segmentation, classification and pose estimation. Recognising object from a single view data has been extensively studied (Russakovsky et al., 2015), however, in many circumstances, single view information is not sufficient to reliably recognise the object. One way to increase the recognition accuracy is to increase the quality of the sensor (*e.g.* increasing resolution) or to add different modality (*e.g.* depth map). However, single view data is still limited by FOV, lighting and particularly, occlusion. Multiple view of the object can help improve the recognition performance (Collet and Srinivasa, 2010) and view planning for object recognition can actively select the most discriminative view of the object for the task. View planning for object recognition is different from 3D reconstruction since the desirable views for object recognition are the view with the most salient feature visible, whereas

3D reconstruction targets to scan the whole object with maximum coverage. Shapenet (Wu et al., 2015) have proposed a deep representation for volumetric shapes for shape recovery and classification as well as the next best view prediction. Shapenet represented a geometric 3D shape as a probabilistic distribution of binary variable on voxels grid and formulated the recognition uncertainty as entropy of the class variable conditioned on the observed voxel. The algorithm predicts the new observable voxel in the next view candidate using ray tracing and predicts the change in entropy. The view candidate with the maximum entropy reduction is selected as the next-best-view. Cheng, Wang and Fragkiadaki (2018) have proposed a 3D latent feature tensor with a recurrent DNN to accumulate features from multiple views for the task of segmentation and classification. The view policy network directly estimates the next view from eight candidate views with REINFORCE (Sutton et al., 2000) algorithm. Kiciroglu et al. (2020) have proposed an active system for human body pose estimation with a camera mounted on a drone. The body pose estimation is modelled with Gaussian distribution where the uncertainty is measured as the sum of the eigenvalues. Future pose in the next time step is forecasted along with the measurement error to select the view candidate with the least uncertainty.

2.3.4 *Sensor motion planning*

Sensor motion planning stage generates possible robot trajectory to move the sensor to the view selected by the view planning stage. Examples of path planning algorithm include Rapidly-exploring Random Tree or its variant RRT* as used in Bircher et al. (2016) for mobile robots. Inverse-kinematic solvers are used for a sensor mounted on robotic arms. Often motion planning is used before view planning to sample reachable view candidates. In the

literature, sensor motion planning is often omitted by assuming every possible viewpoints are reachable (Cheng, Agarwal and Fragkiadaki, 2018) or fix a set of viewpoints (Cheng, Wang and Fragkiadaki, 2018; Wu et al., 2015).

This section introduced the general pipeline for active vision system and summarised approaches used for different tasks. Most robotic active vision approaches generate a set of candidate views based on the robot hardware or task settings from which the next-best view is selected. Approaches with more theoretical analysis with synthetic data often either assumes all possible views are accessible or only a fixed set of views are available. The strategy to select the next best view differs depending on the task: For reconstruction tasks, the view planning module encourages to explore unknown information in general whereas recognition tasks focuses more on saliency. In addition to the information gain, many approaches considers cost associated with the action such as movement costs.

2.4 DOMAIN ADAPTATION

This section presents a general literature review on visual domain adaptation for DNN. Csurka (2017) has categorised domain adaption based on the different approaches to reduce the domain gap: Discrepancy-based, Adversarial-based and reconstruction-based. Each category can be sub-divided to supervised and unsupervised depending on whether annotated data is available in the target domain, but only unsupervised DA will be discussed in this thesis.

DISCREPANCY-BASED DA Discrepancy-based DA assumes that the model trained with data and label in the source domain can be fine-tuned with labelled/unlabelled data in the target domain to diminish the domain shift. Different criteria are used

to minimise the discrepancy between models trained with source domain and target domain. When no annotated data is available in the target domain, the most commonly used approach is to use Maximum Mean Discrepancy (MMD) (Hu et al., 2015; Long et al., 2016; Yan et al., 2017). MMD is a non-parametric metric for computing the distance between distributions and it can be used as loss function to reduce the difference between the two domains in the feature space. MMD helps align the learned features for both source and target domain, but there is no guarantee the nearby samples of target domain features have the same class label in the embedding space (Motiian et al., 2017).

ADVERSARIAL-BASED If a training dataset contains samples from more than one domain, each sample can be easily labelled with which domain the sample is drawn from. This domain class can be exploited to guide the model to learn a domain-invariant classifier. Domain discriminators which classify whether the input data is drawn from the source domain or the target domain are used to encourage domain confusion through an adversarial objective. Two different approaches fall into this category: generative and non-generative.

Generative approach alternately trains discriminator and generator to learn target domain distribution. Using the learned generator, synthetic images with labels in the target domain can be generated. CoGAN (Liu and Tuzel, 2016) has trained two pairs of generators and discriminators, one for each domain with weight sharing. The trained model can generate paired images that are from the source and the target domain, which can be used to train a model for the target domain. SBADA-GAN (Russo et al., 2018) has trained bi-directional image transformations. Proposed class consistency loss imposes that, by transforming source image to the target domain and then again to the source domain,

they should get back to their ground truth class.

Non-generative approach aims to learn domain invariant representation directly from the domain label, instead of learning to generate fake images. Domain-Adversarial Neural Network (DANN) (Ganin et al., 2016) has proposed Gradient Reversal Layer (GRL) that reverses the gradient from domain classifier. The Architecture is designed to have a shared feature extractor with two branches: domain classifier with GRL and task label predictor. DANN minimises both task loss and domain classifier loss. Since the gradient from domain classifier is reversed through GRL, it maximises the domain confusion. Tzeng et al. (2017) have claimed that the generative approach can generate fake images with compelling visual but not optimal for discriminative tasks. The authors proposed Adversarial Discriminative Domain Adaptation (ADDA). Unlike DANN, ADDA learns separate feature extractor for source domain and target domain, which leads to learning more domain-specific feature.

RECONSTRUCTION-BASED Reconstruction based approaches achieve domain invariant feature through reconstruction task. Bousmalis et al. (2016) have proposed Domain Separation Networks (DSN) which explicitly learn to extract two separate image representation: one that is shared across domains and one that is specific to each domain. A shared decoder learns to reconstruct the input image that is sampled from both the source and the target domain. Similiary, Ghifary et al. (2016) have proposed Deep Reconstruction-Classification Networks (DRCN) which jointly learns supervised source label prediction and the unsupervised target data reconstruction. By alternately training the model with the two tasks, a commonality between the tasks are learned in the shared representation. DiscoGAN (Kim et al., 2017) and CycleGAN (Zhu et al., 2017) have used dual learning technique

first proposed by He et al. (2016) for machine translation. Dual learning learns the distribution mapping (translator) between two domains. Cyclic consistency is imposed by translating image in the source domain to the target domain then reconstruct to the source domain.

This section introduced and categorised different approaches in the domain of deep visual domain adaptation. Although deep domain adaptation has been used successfully recently, mostly existing algorithms address the classification task. How to achieve domain adaptation on other vision tasks is probably on e of the main challenges that should be addressed.

3

SINGLE-IMAGE 6D OBJECT POSE ESTIMATION

3.1 OVERVIEW

Previous works on 6D object pose estimation mostly focused on the scenario where Single Instance of Single Object (SiSo) is in the scene with no or low level of occlusion. Recently, Benchmark for 6D Object Pose Estimation (BOP) challenge (Walas et al., 2017) brought large number of existing 6D pose estimation methods to be analysed with the same set of dataset and criteria. BOP also evaluated under SiSo setting. Such setting simplifies the problem to 'finding the instance in the most favourable pose with least occlusion and unambiguous view'. However, some applications, such as AR or robotic grasping, require to estimate multiple objects simultaneously often with severe occlusion. For instance, robotic grasping in an industrial setting can take place in an environment where multiple objects are contained in a box as demonstrated in Amazon Picking Challenge (APC) (Eppner et al., 2016). The challenge involves the task of moving target items from the shelf into a tote where each bin contain between one and 10 items. The shelf may contain multiple copies of the same item. In such case, the agent needs to estimate the poses of the multiple objects and select the object with the pose which is the easiest to grasp with the robotic gripper. When objects are randomly stacked in a box, or on a pallet, the instances are inher-

ently subjected to severe occlusion and clutter, thus estimating the 6D poses of multiple instances is challenging.
Prior-arts for 6D object pose include template-based methods (Hinterstoisser et al., 2011), point-pair features (Drost et al., 2010), or per-pixel regression/patch-based on top of Hough forests (Brachmann et al., 2014; Doumanoglou et al., 2016; Krull et al., 2015; Tejani et al., 2014). These methods have been mostly applied and tested on single object scenario with mild occlusion. More recent methods adopt CNNs for 6D pose estimation, taking RGB images as inputs (Kehl et al., 2017; Rad and Lepetit, 2017; Tekin et al., 2018). BB8 (Rad and Lepetit, 2017) and Tekin et al. (2018) have performed corner-point regression followed by PnP for 6D pose estimation. The methods have shown comparable performance to other RGBD-based methods; however, did not perform well on the occlusion dataset (Brachmann et al., 2014).

For the crowd scenarios, *i.e.*, bin-picking, rather than considering each hypothesis individually, Aldoma et al. (2012) have formulated a joint registration problem. It obtains an optimal set of hypotheses of multiple objects by minimizing the cost function. Doumanoglou et al. (2016) have collected a real dataset of bin-picking scenarios, proposing 6D pose estimator based on a sparse auto-encoder and hough forest, and then tackling the joint registration in the similar way (Aldoma et al., 2012). Recently, Brégier et al. (2017) have proposed a dataset of both synthetic and real in bin-picking scenarios, and Park et al. (2017) have tackled deformable objects in crowd by matching local descriptors.

In the domain of 2D object detection, recent methods have shown that learning the relation between nearby hypotheses in crowd scenario can increase the detection performance by effectively suppressing false hypotheses. Hosang et al. (2017) have proposed a method to learn non-maximum suppression for 2D bounding boxes. More recently, Hu et al. (2018) have trained a 2D detection network jointly with relation module in an end-to-end

manner.

In this chapter, I propose a depth-based 6D pose estimation network which jointly detects and directly recovers full 6D poses of multiple objects in the bin-picking scenario from single depth images. In order to train the proposed network, a training dataset generation framework which produces realistic occlusion pattern is also proposed. Directly estimating multiple 6D poses of objects from a single image is a difficult task, therefore the architecture has multiple modules optimized for different sub-tasks: image 2D detection, depth estimation and 3D pose estimation for individual objects, and joint registration of multiple objects as shown in Figure. 3.1. Although the full architecture consists of several modules, the tasks are jointly learned. The proposed network explicitly learns physically plausible object pose configurations and occlusion patterns from depth images generated by physics simulator and 3D models of objects.

2D offset estimation regresses 2D centre position of the object given the detected box coordinate to compensate misalignment during the bounding box regression. Multiple hypotheses per detected object are generated from the depth and 3D pose estimator by formulating the estimation problem as a classification task. Lastly, relations between hypotheses from the pose estimator are modelled by the joint registration network using block modules (Hosang et al., 2017) to suppress false positives.

3.2 RELATED WORK

In this section, I first review 6D object pose recovery for *crowd scenarios*. Then, I present studies on *relational reasoning*, which is used to improve the confidence of a 6D hypothesis in this chapter.

Crowd scenarios. Bin-picking scenario is a very challenging task for any vision algorithm since it exhibits several cascading issues (e. g., severe occlusions, foreground clutter, etc.). The method proposed by Liu et al. (2012) extracts robust depth edges and employs fast directional chamfer matching to detect objects and estimate their poses reliably. Aldoma et al. (2012) obtained an optimal set of hypotheses of multiple objects by minimizing a cost function, rather than considering each hypothesis individually. Doumanoglou et al. (2016) have proposed 6D pose estimator of individual objects based on a sparse auto-encoder and hough forest tackling the joint registration in a similar way as Aldoma et al. (2012). However, the method is computationally demanding with increasing number of hypotheses. Park et al. (2017) have tackled deformable objects (fruits) in the crowd by matching local descriptors. Recently, Zeng et al. (2017) have addressed the challenges of the crowd scenarios proposing a self-supervised deep learning method for 6D pose estimation in Amazon Picking Challenge.

Relational reasoning. Although most of the modern object detectors generate detection hypotheses independently, recent works have shown exploiting the relation between hypotheses helps under high occlusion. Lee et al. (2016) have modelled the relation between detection hypotheses and used determinantal point process to select the most confident hypothesis. Stewart et al. (2016) have proposed a network for person detection which eliminates the non-maximum suppression stage by modelling the relation between detections using LSTM (Hochreiter and Schmidhuber, 1997). Gossipnet (Hosang et al., 2017) have performed non-maximum suppression by proposing a block module which develops a representation for each detection which accounts for other nearby detections. Hu et al. (2018) have added an extra layer called relation module to the standard object detection framework and jointly trains in an end-to-end fashion. It has shown that

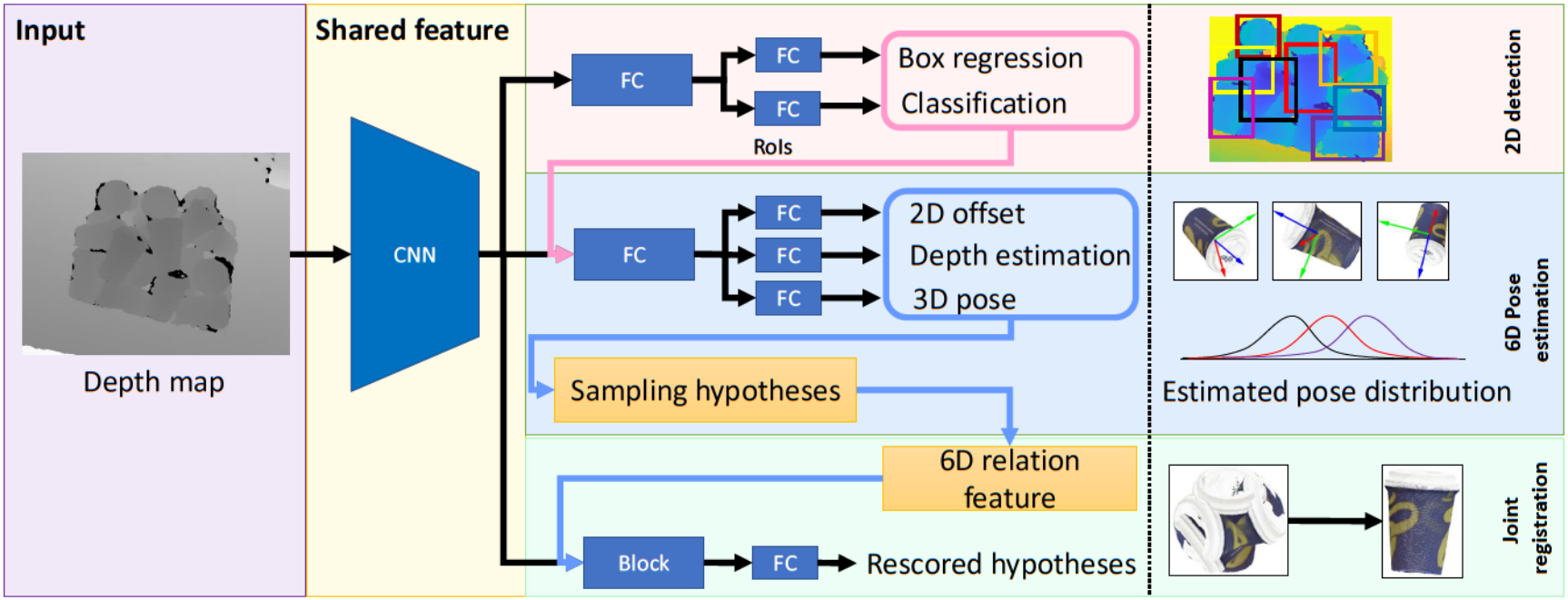


Figure 3.1: The proposed multi-task learning network architecture. It consists of three modules, each optimized for different sub-tasks: 2D detection, 6D pose estimation, and joint registration. During inference, each sub-tasks are sequentially performed (red and blue arrows). During training, the network is trained jointly using multi-task losses, generating the convolutional features shared across sub-tasks.

learning object detector with relational reasoning improves detection performance. Both Hosang et al. (2017) and Hu et al. (2018) have used relational reasoning to remove duplicate hypotheses in 2D detection. In this work, the architecture developed in Hosang et al. (2017) is extended to 6D pose estimation problem.

3.3 PROPOSED METHOD

A key challenge in object pose estimation in bin-packing scenarios is that multiple object instances exhibit severe occlusions under environments cluttered with the same object. For severely occluded objects, the 6D pose predictions made based on visible parts are inherently ambiguous (see Figure. 3.4). Therefore, a joint registration step exploiting the neighbourhood context (of how the pose of an object is related to neighbouring objects) can help make predictions reliable. Existing approaches address this issue only at the testing stage, e. g., using non-maximum suppression (NMS) and the training of their pose estimators is agnostic to such occlusion behaviours: They are trained on isolated object instances. Instead, the proposed directly incorporates the variety of occlusion patterns and the possibility of exploiting the neighbourhood context into the *training* phase. To facilitate this process, a new training dataset is constructed that reflects such a realistic occlusion-heavy application environment.

Figure 3.1 shows the proposed pose estimator architecture consisting of three modules, each instantiated as a DNN: The detection module (red) estimates 2D (along the image plane in the depth map) object locations as bounding boxes. Within each hypothesized bounding box, 6D pose estimation module (blue) estimates the centre of the object (which may differ from the centre of its bounding box) and its depth (completing the 3D locations), and based on them, generates multiple 3D pose hypotheses (as singling out a correct pose is challenging under

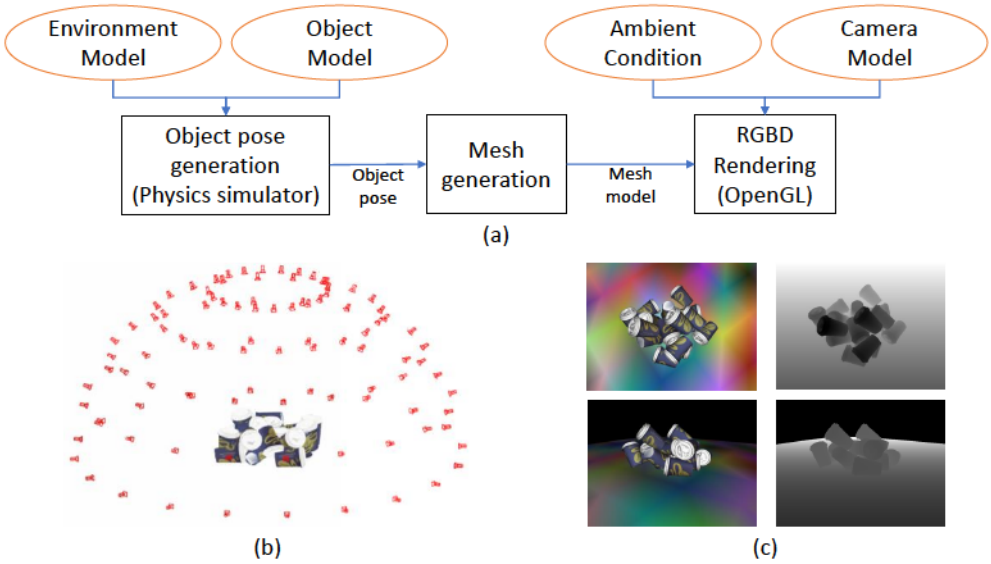


Figure 3.2: Training data generation process: (a) A realistic pose configuration is generated via physical simulation (b) 100 views are sampled on the upper hemisphere (c) Example training views rendered from the scene

heavy occlusions). Finally, the joint registration module (green) disambiguate between multiple 6D pose hypotheses and eliminate false positives by considering local neighbourhood structures. Exploiting the dependency present in these three tasks and the corresponding modules, I formulate their joint training as multi-task learning (MTL): All three modules share the same depth feature extractor (yellow block in Figure. 3.1), enabling the training of convolutional features in a unified manner.

The overall loss function is defined per depth image as:

$$L = L_D + L_O + L_D + L_P, \tag{3.1}$$

with L_D, L_O, L_D, and L_P being the losses for estimating 2D bounding boxes, 2D object centres, 3D depths, and 6D poses, respectively. During training, the backward propagation takes place as usual for each module, and they are subsequently combined to update the shared feature extractor in Figure. 3.1. The 2D detection and 6D pose estimation modules are simultaneously

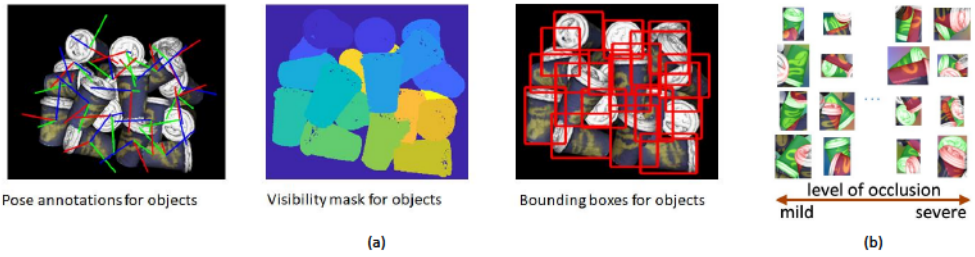


Figure 3.3: Training dataset generation output: (a) pose, segmentation mask, bounding box annotations are automatically generated. (b) Example occlusion patterns with varying degree of occlusion.

updated, while for the update of the joint registration module, I hold the rest of the network weights fixed for improved convergence. The remainder of this section details each module and the corresponding individual loss function in Eq. 3.1. I have experimented with different weights applied for each loss terms, however changing the weights did not affect the overall performance largely. Therefore I have fixed the weight of each term equal to 1.

GENERATING TRAINING IMAGES. Unlike most existing 6D pose estimators which learn object appearance from *isolated views* of objects, the proposed algorithm jointly learns object appearances and their mutual occlusion patterns based on cluttered scene images. Figure 3.2(a) illustrates the generation of training data in this scenario: First, the cluttered scene in a virtual environment is constructed by randomly stacking the object of interest in a box. Each object has 2,000 different bin picking scenes, each of which has 10 to 20 instances in the virtual bin. Then, training depth images are rendered by sampling virtual camera views (Figure. 3.2(b)): The camera is always facing towards the centre of the bin but with randomized in-plane rotations to capture various occlusion patterns. Seventeen depth images are generated from each scene, constituting a set of 34K images per object.

Groundtruth labels for each objects in the image are automatically generated as shown in Figure. 3.3

2D BOUNDING BOX DETECTION. In a bin-picking scenario, the number of objects in a given scene is not known a priori. Therefore, the traditional detection strategy of predicting the single most probable location for SiSo is not applicable. I employ Faster RCNN (Ren et al., 2015) which has shown its effectiveness in detecting multiple objects in cluttered scenes. While Faster RCNN was originally developed for RGB images, I have modified the input layer of Faster RCNN to take a single channel depth map instead of the three channel RGB image. The network receives a single image and generates a bounding box and its confidence per object instance: During training, it minimizes the box regression error (via L_1-loss), and for the detection confidence, the soft classification error (via cross entropy) of foreground and background. The combination of the two losses are denoted as L_D (see Eq. 3.1).

6D POSE ESTIMATION. The 6D pose of an object consists of its 3D location and 3D pose (orientation). Given the bounding box of each object instance estimated by 2D detection module, 6D pose estimation module generates multiple hypothesized 6D poses, which are subsequently fed to the joint registration module.

2D object centre estimation. For each bounding box, the accurate 2D location of the corresponding object centre is estimated first. In existing 6D pose estimation approaches (developed for uncluttered scenes) (Do et al., 2018; Kehl et al., 2017), the centre of an estimated box is considered identical to the centre of the object it contains. This is reasonable only for uncluttered scenes, as in such scenario, the estimated boxes are well-aligned with the underlying tight ground-truth bounding boxes. However, I experimentally observed that for objects under severe occlusion, such

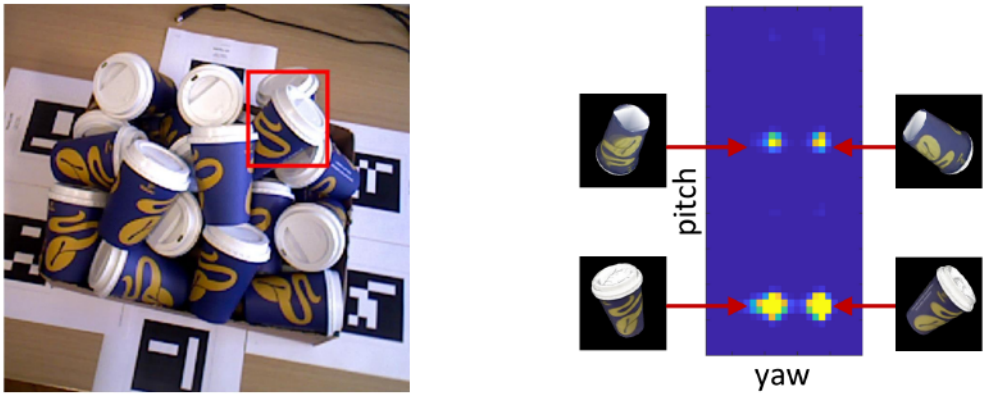


Figure 3.4: Multiple hypotheses generation: (a) An example detection under severe occlusion (overlaid on the RGB image for visualization);(b) Multiple hypotheses with confidences (the brighter is the higher in the heatmap) generated by the 6D pose estimator. The bottom two pose hypotheses correspond to the two cups that are visible in the red bounding box in (a).

accurate bounding box estimation is extremely challenging, and naïvely basing the joint registration on such erroneous 2D location estimates eventually leads to erroneous 6D pose predictions (see Table 3.1(a) 'Ours (w/o offset)'). Therefore, an additional 2D regression network is constructed that explicitly estimate the object centres by minimizing the deviation from the ground-truth within each bounding box:

$$L_O = \left\| \left[\frac{x - c_x}{w}, \frac{y - c_y}{h} \right]^\top - \left[\frac{x^* - c_x}{w}, \frac{y^* - c_y}{h} \right]^\top \right\|_1, \tag{3.2}$$

where (x, y) and (x^*, y^*) denote the predicted and the ground-truth object centre coordinates, respectively, (w, h) represents the width and height of the detected box, and (c_x, c_y) is the top-left corner coordinate of the box.

3D pose and depth estimation. A common approach to 3D pose estimation is regression, e. g., to estimate the quaternion (Alex and Roberto, 2017) or the elements of rotation matrices (Do et al., 2018). However, in the preliminary experiments, it has been

observed that when the object undergoes severe occlusion, generating a single reliable pose estimate is challenging: Inspected at only small portions of objects, indeed, significantly different pose configurations can be instantiated from visually very similar depth values (see Figure. 3.4 for example). To circumvent this challenge, I first generate multiple pose hypotheses and disambiguate them afterward by exploiting the context of neighbouring detections and their pose hypotheses during joint registration. To facilitate hypothesis generation, I adopt the classification approach of Kehl et al. (2017): Three rotation components are estimated separately, each represented as a classification problem with different classes binning an angle range: I use 30, 13, and 30 classes for pitch, yaw, and roll, respectively. For objects with axial symmetry, the roll component is not estimated. Similarly, the 3D depth values are estimated via classification (into 140 bins). The standard soft-max cross-entropy loss is used for training (L_P for 3D pose and L_D for depth):

$$L_{P(D)} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{y}_i^* \cdot \log(\mathbf{y}_i), \tag{3.3}$$

where $\mathbf{y}^*$ is the ground-truth in the form of one-hot encoding, $\mathbf{y}_i$ is the class estimation after softmax activation, and N is the batch size. This formulation allows to generate multiple hypothesis each with confidence as a product of individual classification confidence for each detection. These confidences are used by the joint registration module along with the spatial configurations of neighbouring detections. For computational efficiency, before joint registration, I sample only five hypotheses per object by first removing many similar hypotheses using NMS, and then selecting the remaining hypotheses with the five highest confidences. The sampling strategy was necessary since using all the

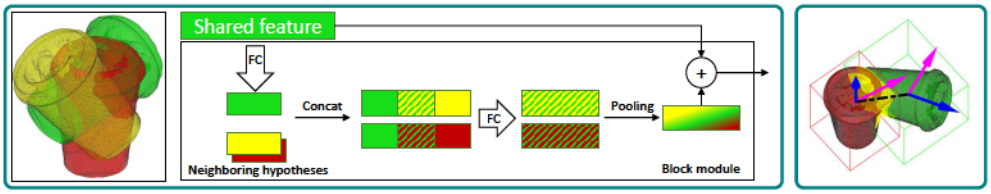


Figure 3.5: Visualisation of joint registration module (a) Schematic diagram of the feature extractor in the joint registration module applied to a hypothesis (green) with multiple neighbouring hypotheses. Hatched boxes are the 6D relation features. Colour in the box corresponds to hypotheses on the left image. Features from different hypothesis and 6D relation feature are concatenated. Variable number of neighbours for each hypothesis is reduced to a single feature by max-pooling. (b) An example neighbourhood pattern: The intersection of the corresponding bounding boxes is highlighted in yellow; The black dotted line joints the centre of bounding boxes; Blue and magenta arrows represent yaws and pitches, respectively.

hypotheses would result in joint registration module to process every possible hypothesis which is 11700 hypotheses.

JOINT REGISTRATION. By design, the pool of hypotheses generated from 6D pose estimation module contains multiple false positives. As no prior information on the number of ground-truth object instances or their spatial configurations is available, the joint registration module needs to consider the spatial configurations of multiple hypothesized object poses simultaneously. I adapt GossipNet framework proposed by Hosang et al. (2017) for this purpose: Applied to the bin-picking scenario, this GossipNet adaptation classifies each hypothesis of interest (HOI) into True positive (1) or False positive (-1) classes. The classification is made based on a *contextual feature* encapsulating the spatial configurations of neighbouring hypotheses (whose 3D bounding boxes overlap with the HOI). Figure 3.5(a) visualises this feature extraction process: First, for a given HOI (green), all neighbouring hypotheses (NH; e. g.red) are identified. Then, for each identified

NH and HOI pair, an intermediate feature vector is generated by concatenating 1) the image depth features calculated within the bounding boxes of HOI and NH, respectively (solid green and red boxes in Figure. 3.5(a); See Figure. 3.1(shared feature)) and 2) 13-dimensional feature vector of their joint configurations(6D relation feature): (1-2) confidence score of each detection, (3) intersection over union score of bounding boxes, (4-9) Euler angles normalised to $[0, 1]$, (10-12) distances between the two detection centres along each x, y, and z direction, normalised by the bounding box diameter, and (13) normalised L_2 distance between the two detection centres.

The multiple intermediate pair-wise features (for all identified NHs) constructed in this way are then pooled to make a single contextual feature vector facilitating the classification independently of the number of neighbouring hypotheses. During training, the registration network f_J (combining the contextual feature extractor and a fully connected network classifier) minimises the classification loss given below:

$$L_J(f_J) = \sum_{i=1}^{N} a_i \cdot \log(1 + \exp(-f(h_i) \cdot y_i)). \tag{3.4}$$

Here, the batch-size N is the number of total hypothesis $\{h_i\}$ in a single depth image, $y_i \in \{-1, 1\}$ represents the ground-truths, and a_i is a weight compensating the class imbalance decided at 1 and 16 for 1 and -1 classes, respectively.

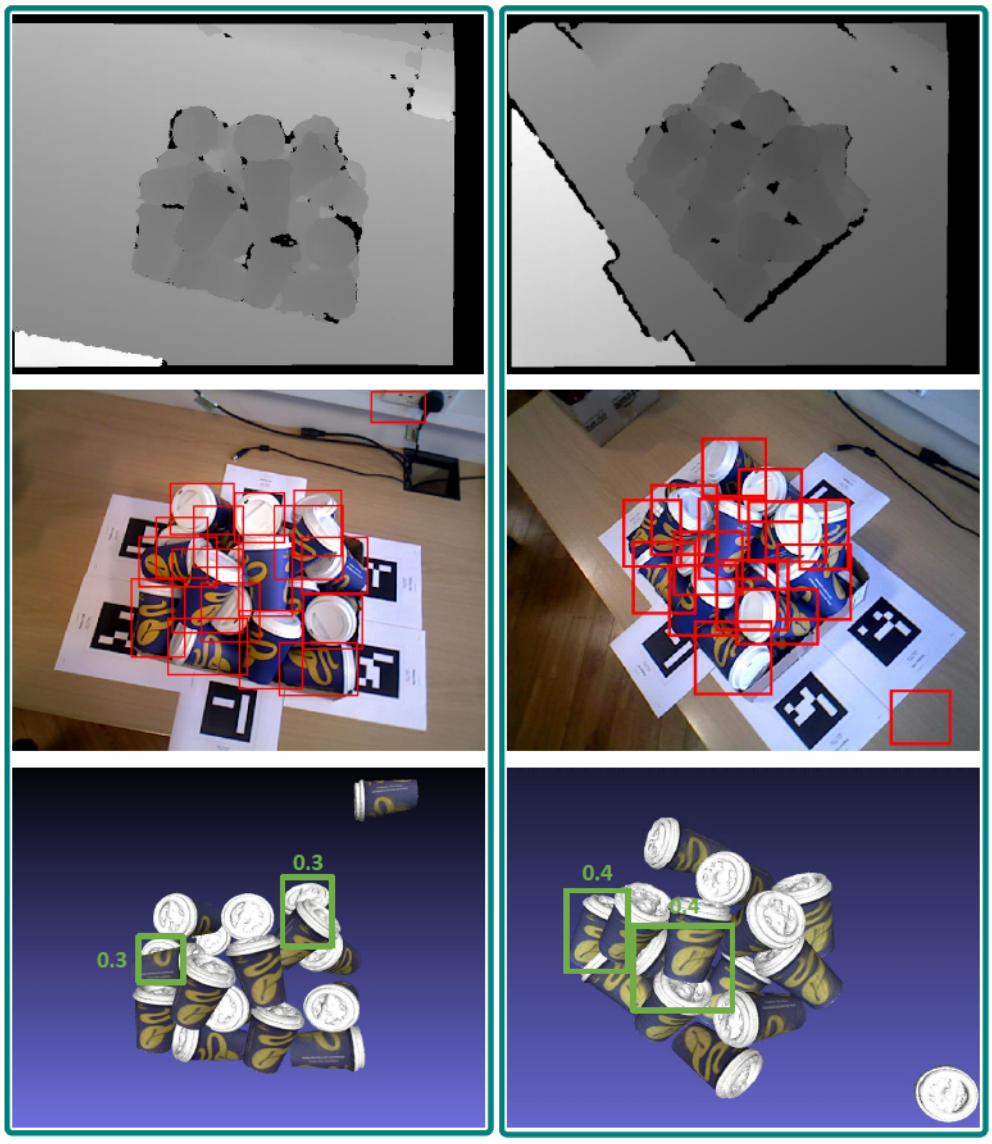


Figure 3.6: Qualitative results of top 20 hypotheses for the *Coffee cup* scenario. (top) Input depth image. (middle) Detected bounding boxes. RGB image is used for visualization. (bottom) Reconstructed scene from hypotheses. Difficult objects are highlighted in green box annotated with visibility ratio. Objects with severe occlusion, some of which have visibility as low as 30%, are successfully recovered.

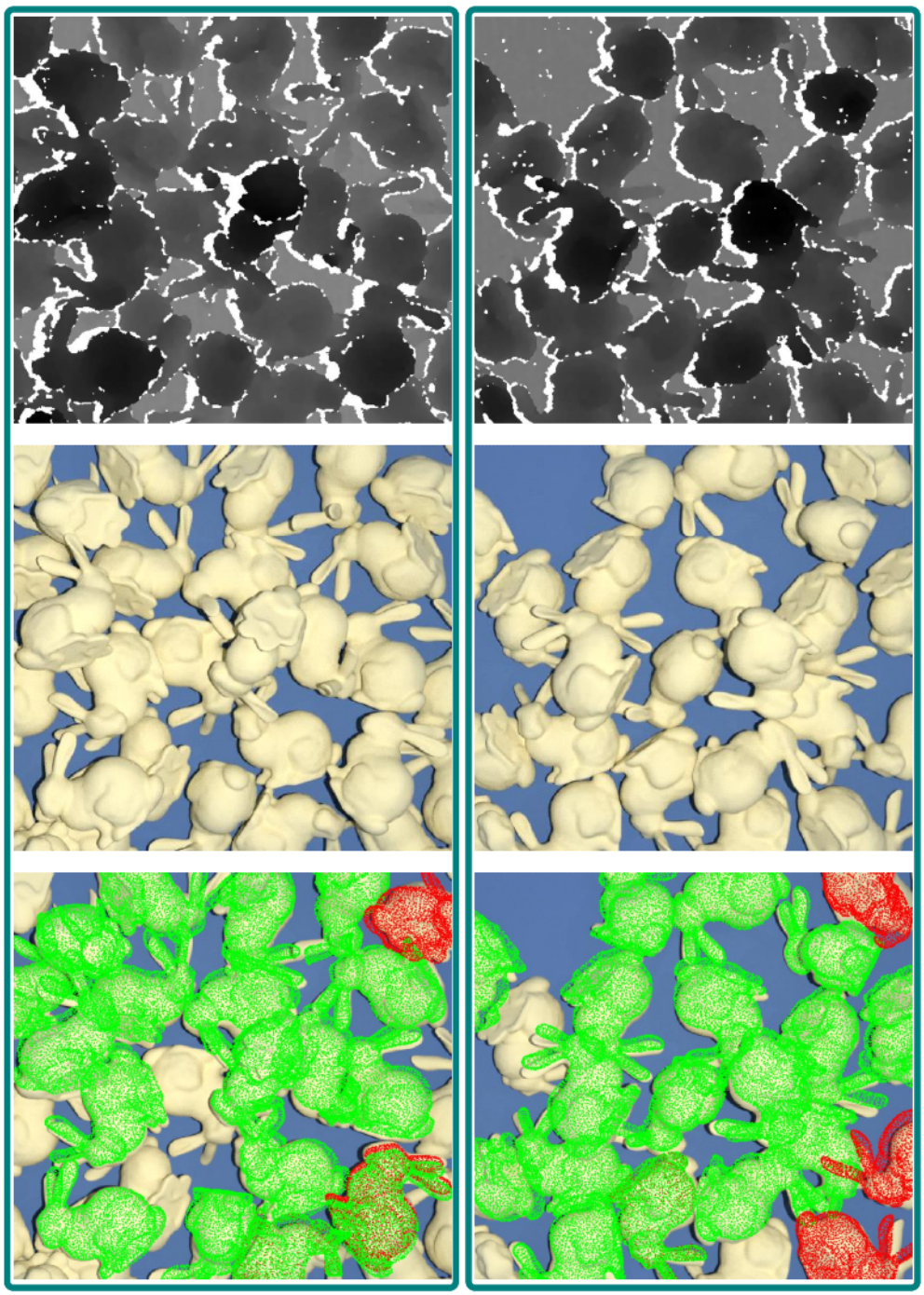


Figure 3.7: Qualitative results of top 20 hypotheses for the *bunny* scenario. For each triplet: (top) and (middle) The input depth image and the corresponding RGB image (for visualization); (bottom) Hypotheses projected onto the RGB image. True and false positives are highlighted in green and red, respectively

3.4 EXPERIMENTAL EVALUATION

The proposed algorithm is evaluated on four objects from two different bin-picking datasets where multiple object instances occlude each other, reflecting realistic application cases: *Coffee cup* from the Bin-Picking dataset (Doumanoglou et al., 2016), and *Bunny*, *Pepper*, and *Candlestick* from the Siléane dataset (Brégier et al., 2017). Qualitative results for *Coffee cup* and *Bunny* are visualised in Figure. 3.6 and Figure. 3.7. The proposed method is compared with existing algorithms: Linemod algorithm (Hinterstoisser et al., 2011); 3D point-pair-feature (PPF)-based algorithm (Drost et al., 2010); and Latent-class Hough forest (LCHF) (Tejani et al., 2014). While PPF and LCHF are known to be robust to scene clutters and occlusions, respectively, they are not specifically designed for bin-picking applications. Therefore, I also compare with hypotheses verification approach (Aldoma et al., 2012) that extends Linemod and PPF for bin-picking applications (referred to as 'Linemod+PP' and 'PPF+PP', respectively). To gain an insight into the effectiveness of the proposed multi-task learning approach in general 6D pose estimation (non bin-picking cases), the proposed algorithm is evaluated on the Domestic Environment dataset (*Coffee cup* object) (Tejani et al., 2014) providing examples of multiple object instances under mild occlusions.

The accuracy of predictions is measured in average precision (*AP*; equivalently, the area under the precision-recall curve) evaluated based on *Sym* criteria adopted from Brégier et al. (2017): Based on *Sym*, a hypothesis is accepted (considered as a true positive) if the distance to ground-truth is smaller than 10% of the diameter of the smallest bounding sphere. I calculated the AP values of the proposed algorithm using the evaluation code shared by the authors of Brégier et al. (2017) enabling a direct comparison with the results of Linemod(+PP) and PPF(+PP) supplied from Brégier et al. (2017). I also measure the AP based on

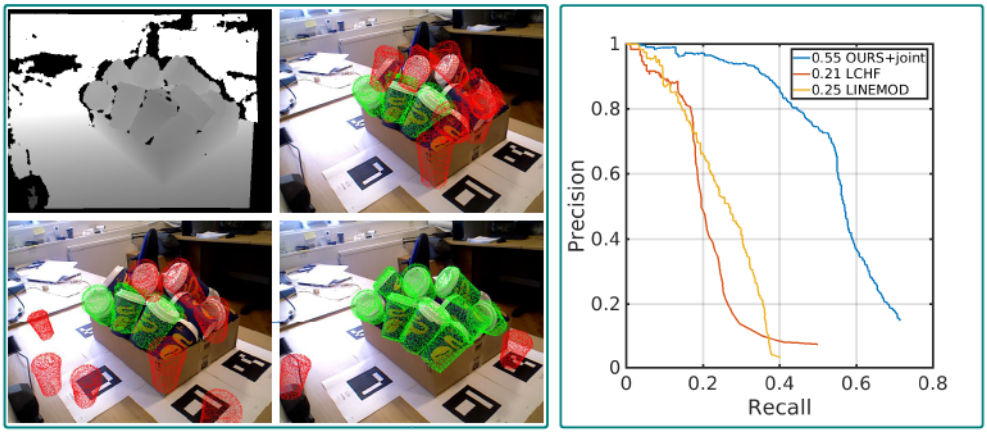


Figure 3.8: Comparison with other methods. (a) Input depth image and example 6D pose estimates by Linemod, Ours, and LCHF, respectively (clockwise from top-left). Correctly estimated poses are highlighted in green. (b) The precision and recall curves of Linemod, LCHF, and the proposed algorithm on *Coffee cup* dataset.

ADD criteria (Hinterstoisser et al., 2012). *ADD* is the average distance of all model points between the groundtruth pose and the estimated pose. It is calculated as

$$m = \underset{\mathbf{x}_1 \in \mathcal{M}}{\text{avg}} \min_{\mathbf{x}_2 \in \mathcal{M}} \left\| (\mathbf{R}\mathbf{x}_1 + T) - (\hat{\mathbf{R}}\mathbf{x}_2 + \hat{T}) \right\|, \tag{3.5}$$

where $\mathcal{M}$ is the 3D object model, R, T and $\hat{R}, \hat{T}$ are groundtruth object pose and estimated pose respectively. A hypothesis is accepted if its average distance, m, is less than 0.1 times the object diameter.

RESULTS ON BIN-PICKING DATASETS. [1] Table 3.1(a) summarizes the AP scores. No clear winner was found among the two classical algorithms, Linemod and PPF: Each one is better than the other on two datasets. As these algorithms are not designed for heavily cluttered-scenes, applying extra verification (Aldoma

1 As the original Bin-Picking dataset contained erroneous ground-truth annotation, every frame has been manually re-annotated via visual inspection.

Table 3.1: Quantitative results. **(a-b)** AP values of different algorithms in crowd scenario: Each pose hypothesis is evaluated in *Sym* criterion **(a)** and *ADD* criterion **(b)**. **(c)** F1 scores for Domestic Environment dataset (*Cup* object) (Tejani et al., 2014) with *ADD*.

Method	*Bunny*	*cup*	*pepper*	*C.stick*
Linemod (Hinterstoisser et al., 2011)	0.39	0.08	0.04	0.38
Linemod+PP (Aldoma et al., 2012)	0.45	0.20	0.03	0.49
PPF (Drost et al., 2010)	0.29	0.28	0.06	0.16
PPF+PP (Aldoma et al., 2012)	0.37	0.30	0.12	0.22
Ours (w/o offset)	0.49	0.53	0.32	0.53
Ours	**0.74**	**0.55**	**0.43**	**0.64**

(a)

Method	Cup (real)	Cup (synt.)
Linemod (Hinterstoisser et al., 2011)	0.38	0.32
LCHF (Tejani et al., 2014)	0.30	0.31
SSD6D (Kehl et al., 2017)	0.07	0.01
Ours	**0.68**	**0.80**

(b)

Method	Cup (Tejani et al., 2014)
Linemod (Hinterstoisser et al., 2011)	0.819
PPF (Drost et al., 2010)	0.867
LCHF (Tejani et al., 2014)	0.877
Sahin et al. (2017)	0.793
Kehl et al. (2017)	0.972
Ours	**0.992**

(c)

et al., 2012) significantly improved performance (Linemod+PP and PPF+PP). None of these four existing methods accounts for the occlusion patterns during training. Instead, they use the ICP to verify and refine the estimated hypothesis. By explicitly and simultaneously learning the occlusion patterns and object shapes during training, the proposed algorithm (Ours) constantly outperformed existing algorithm with a large margin (15-31%) even without having to rely on the external ICP algorithm.

For qualitative comparison, Figure. 3.8(a) shows example detections: LCHF, Linemod, and Ours all correctly estimated the 6D poses of unoccluded foreground objects. However, LCHF and Linemod missed multiple objects with only mild levels of occlusions (highlighted in red), while the proposed algorithm correctly recovered all objects with only two false positives leading to constantly and significantly higher precision across different recall values (Figure. 3.8(b)).

Due to severe occlusions caused by multiple instances of a single object, estimating the 6D poses in bin-picking using pure RGB-based approaches is challenging. I verified this by applying the state-of-the-art SSD6D algorithm (Kehl et al., 2017) in this scenario[2] (Table 3.1): Not surprisingly, the results of SSD6D algorithm failed to match the level of any depth image-based pose estimation algorithms in comparison.

'Ours (w/o offset)' in Table 3.1(a) shows the AP score of the proposed algorithm obtained without the object centre estimation to assess its contribution: Under severe occlusion, the initial bounding box detections are not tightly aligned with the ground-truths. Therefore, excluding this component severely degraded the joint registration process.

I also evaluate the performance of the proposed algorithm with *ADD* (Table 3.1(b)). Note that both Linemod and LCHF,

2 I used the code kindly shared by the authors of this work.

in comparison, use RGBD data whereas Ours uses only depth images.

NON BIN-PICKING SCENARIO. I use the F1 score following publicized results supplied in (Kehl et al., 2017; Tejani et al., 2018) (Table 3.1(c)): Even though the proposed algorithm was trained for bin-packing scenes (with training data consisting of only foreground scenes), and it does not use the ICP refinement in contrast to others, it significantly outperformed Linemod (Hinterstoisser et al., 2011), PPF (Drost et al., 2010), and LCHF (Sahin et al., 2017) and is on par with the state-of-the-art SSD6D (Kehl et al., 2017) demonstrating the effectiveness and robustness (to unseen background clutters) of the proposed algorithm.

IMPLEMENTATION DETAILS. The shared feature extractor (Figure. 3.1) is initialised with Imagenet-trained VGG network (Russakovsky et al., 2015). The entire network was trained using the ADAM optimizer with 10^{-5} learning rate. Training the network on each object took around 10 hours in a machine with a GTX1080ti GPU. Testing a single depth image took 204ms.

3.5 SUMMARY

Object detection and pose estimation in bin-packing scenarios is challenging due to severe occlusions caused by multiple instances of the same object. Existing approaches are limited in that their estimators are trained on isolated object instances and they do not directly consider possible mutual occlusion patterns during training. I presented a new 6D pose estimation framework that is trained directly on the scenes demonstrating multiple occlusion patterns. The proposed framework is instantiated as 2D detection, 6D pose hypothesis generation, and joint registration submodules which are jointly trained via a new multi-task loss. The resulting

algorithm outperformed the state-of-the-art 6D object pose estimation approaches on the challenging datasets. The proposed approach has several weaknesses. The first weakness is due to the input type which is the depth map. The trained model will not be able to adapt to depth images captured with different intrinsic parameters. The second weakness is that the experiments were designed for the scenario where all the instances in the scene are from the same object.

4

LONG-TERM MOTION PLANNING FOR OBJECT POSE ESTIMATION

4.1 OVERVIEW

In the last chapter, I tackled the problem of estimating multiple object poses from a single image. However, often the information contained in a single image is not sufficient in both quality and quantity. The information is limited by the sensor resolution, noise, distortion as well as the FoV. The problem is more severe for the bin-picking scenario where most of the objects are only partially visible. Such applications and scenarios are naturally framed in an active setting, where either a robot or a human has the capability of moving the camera to different viewpoints as presented in *e.g.* the Amazon Picking Challenge (Eppner et al., 2016; Zeng et al., 2017). Previous work in active 6D object pose estimation includes the work of Doumanoglou et al. (2016), which explored the problem of estimating the next best camera viewpoint by aiming to reduce the uncertainty of the pose estimator in terms of entropy reduction. However, Doumanoglou et al. (2016) present two limitations. First, the framework is short-sighted, meaning that they move the camera to the next best view independently of the past and future viewpoints and potentially reaching a redundant view. Second, it requires rendering different views before making a camera movement, which is a limitation

on real-world applications where the number of possible viewpoints is high or even infinite. This chapter proposes a framework to tackle the active 6D object pose estimation problem that overcomes the limitations of the previous work. Given a set of objects hypotheses inferred by an object pose estimator, the proposed framework first fuses the information taking into account the previous estimations and chooses the hypotheses that best explain the scene without needing ground-truth information with the use of a verification score. The result of this fusion function is given to an agent, which analyzes all the hypotheses. Using an attention mechanism, the agent directs its attention to the object which it is most uncertain about and moving the camera towards such an object. The agent is trained in simulated scenarios on a reinforcement learning framework, attempting to learn a policy that moves the camera producing the most accurate object poses hypotheses for a given temporal and spatial budget. The sequential nature of the reinforcement learning framework gives the agent the ability to reason temporally and make long term decisions based on previous movements. Furthermore, careful state space design allows the agent to perform inference without the need for rendering the entire viewpoints at each time step. The proposed framework is evaluated on challenging synthetic and real scenarios, showing that the proposed framework can produce camera movements that achieve robust 6D object pose estimation over three baselines: the random baseline which samples random actions from uniform distribution; unidirectional baseline which moves the camera in one horizontal direction around the scene and the maximum distance baseline which selects viewpoints with the longest distance from the previous viewpoints.

4.2 RELATED WORK

4.2.1 *Multi-view object recognition*

Multi-view object recognition system aims to increase the recognition performance by combining information captured from different views. Multiple views of the scene can be used to overcome some critical issues of occlusion, if the view information is used appropriately. Multi-view approach has shown to increase object detection and recognition performance. Coates and Ng (2010) have executed a single shot-based detector on 2D image for multiple views and determined the correspondences to combine the posterior probability detection. Mustafa et al. (2013) have used multiple fixed Kinect and stereo camera to recognize a single object reliably using texlet. Susanto et al. (2012) have detected multiple objects with multiple Kinect using Deformable Part Model (DPM) and used Viewpoint Feature Histogram (VFH) to eliminate false hypotheses.

4.2.2 *Active vision, object detection and poses.*

Several active object recognition systems have been proposed in the recent years. In Ammirato et al. (2017), an active vision dataset and a reinforcement learning based baseline have been proposed to explore the environment to detect an object. D. Jayaraman and K. Grauman. (2016) have presented an end-to-end object recognition framework which actively selects the best view for the classification of an object. Cheng, Agarwal and Fragkiadaki (2018) have presented a method to learn a policy for both grasping and viewing jointly. The method uses a simulated environment which detects an object depending on the occlusion rate. Unlike the proposed framework in this chapter, where multiple objects are present, all three methods include one object of inter-

est in the scene. More related to our work, Doumanoglou et al. (2016) have proposed a Hough forest approach and an entropy-based viewpoint selection for pose estimation. An important drawback of this method is that it does not have a mechanism to resolve conflicting next best view selection in the presence of multiple hypotheses.

4.2.3 *Attention to objects.*

Recent works have shown that employing attention mechanism is useful in the presence of multiple objects, distractors, and clutter. Xu et al. (2016) have used the attention module to select discriminative features for the purpose of view selection for object classification given an object. Kim and Canny (2017) have used a pixel-wise attention module for the self-driving system to build more interpretable agent. Devin et al. (2018) have presented a system for object grasping with object-level attention where an attention mechanism is used to select the object to be manipulated on specific tasks. Similarly, in this chapter, an attention mechanism is proposed that makes the agent focus on the most uncertain object.

4.3 PROPOSED METHOD

Problem definition. Given a target stack of n objects, the aim is to infer the 6D pose $x^i \in \mathbb{R}^6$ of every i visible object in the stack, consisting of its 3D location and 3D orientation in the scene. In other words, the objective is to find the set of object hypotheses $\hat{x} = \{\hat{x}^1, ..., \hat{x}^N\}$ that best describes the scene. The problem is formulated in an active setting, where an agent has the ability to navigate through a finite set of viewpoints $\mathcal{V} = \{v^1, ...v^M\}$, where $v^i \in \mathbb{R}^6$ is the 6D camera pose, and $M = |\mathcal{V}|$ is the number of

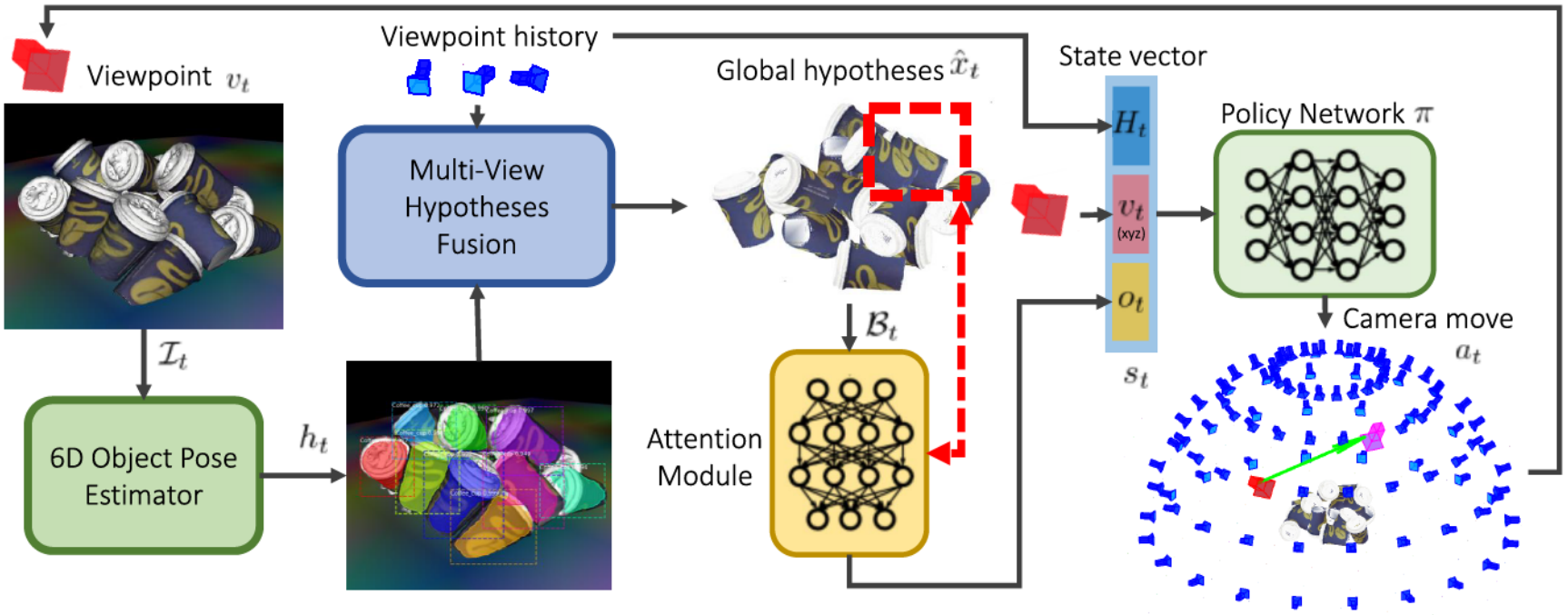


Figure 4.1: The proposed framework. An image from viewpoint v_t is passed through a 6D object pose estimator which generates pose hypotheses. Current and previous hypothesesare accumulated using a fusion function. The agent analyzes the scene, focusing its attention to the object which it is most uncertain about and movingthe camera towards such object.

viewpoints accessible to the agent. At each time step t, the agent observes the state of the scene s_t from a viewpoint $v_t \in \mathcal{V}$ and decides an action a_t following a policy function π that moves the agent to next viewpoint v_{t+1}. The agent proceeds until it reaches the maximum episode length T generating a trajectory $\tau = (s_0, a_0, ..., s_{T-1}, a_{T-1})$. Moving an agent in the real world is costly in terms of time T and energy consumption, which is indirectly measured as a function of total travelled distance d. Both d and T are budget-constrained and low values are desired.

Framework overview. The overview of the framework is presented in Figure. 4.1. At each time step t, an input image $\mathcal{I}_t$ is acquired from a viewpoint v_t and it is passed through a 6D object pose estimator that provides pose hypotheses h_t on the currently observed objects. These hypotheses are accumulated using a fusion function that considers both the current hypotheses and the previously observed ones and a selection of such is given to the agent. In a cluttered scenario, there are multiple objects with different challenges, such as occlusion or measuring pose estimation confidence. This indicates there may not be a single next view ideal for all the objects and thus the agent reasons about the scene and focus its attention to the object which it is most uncertain about and moves the camera towards such object. This process is repeated until the maximum fixed number of time steps T and then the 6D pose hypotheses for the scene x is obtained. In the following sections, I describe the different components of the framework.

4.3.1 *6D object pose estimation*

Given an image $\mathcal{I}_t$ acquired from viewpoint v_t, the 6D object pose estimator outputs a set of object hypotheses for the current view $h_t = \{h_t^1, ..., h_t^k\}$, where $h_t^i \in \mathbb{R}^6$ and k might differ from the actual number of objects in the scene N. The framework is

agnostic to the 6D object pose estimator as long as necessary information for view-planning, detailed in the following section, is provided by the estimator.

4.3.2 *Multi-view hypotheses fusion*

At each viewpoint, v_t, the agent obtains a different set of object hypotheses h_t from the 6D object pose estimator. To obtain a global scene hypothesis $\hat{x}$ that accumulates the information obtained from different viewpoints, I define a hypotheses fusion function f. Given a history of object hypotheses $h_1, ..., h_t$ and camera viewpoints up to time step t, the function f outputs the accumulated hypotheses $\hat{x}_t$ in the global coordinate system. Intuitively, f gathers all the object hypotheses that have been observed so far, selecting the ones that better explain the scene among all the pool of hypotheses, hence $\hat{x}_t \subseteq \bigcup_{i=1}^{t} \bigcup_j h_i^j$. In the framework, f makes use of hierarchical clustering to group hypotheses and the hypothesis with the highest object pose confidence score is selected from each cluster. In the framework, the hypothesis with the highest verification score from each cluster is chosen, as verification score is used to represent quality of the hypothesis. The process of multi-view hypotheses fusion is shown in Figure. 4.2.

Verification score. The hypotheses fusion function requires quantifying the quality of a hypothesis in the absence of ground-truth information. For this, I introduce a verification score that estimates hypothesis confidence inspired by the work of Aldoma et al. (2012) and Doumanoglou et al. (2016). These works assume that a scene point q that belongs to an object -is an *inlier*- if the distance to the nearest neighbouring point p of a rendered 3D model of a hypothesis is lower than a certain threshold ϵ. This assumption is not generally true in the presence of multiple objects in clutter. To overcome this limitation, the segmentation mask is used from the object pose estimator and points on the

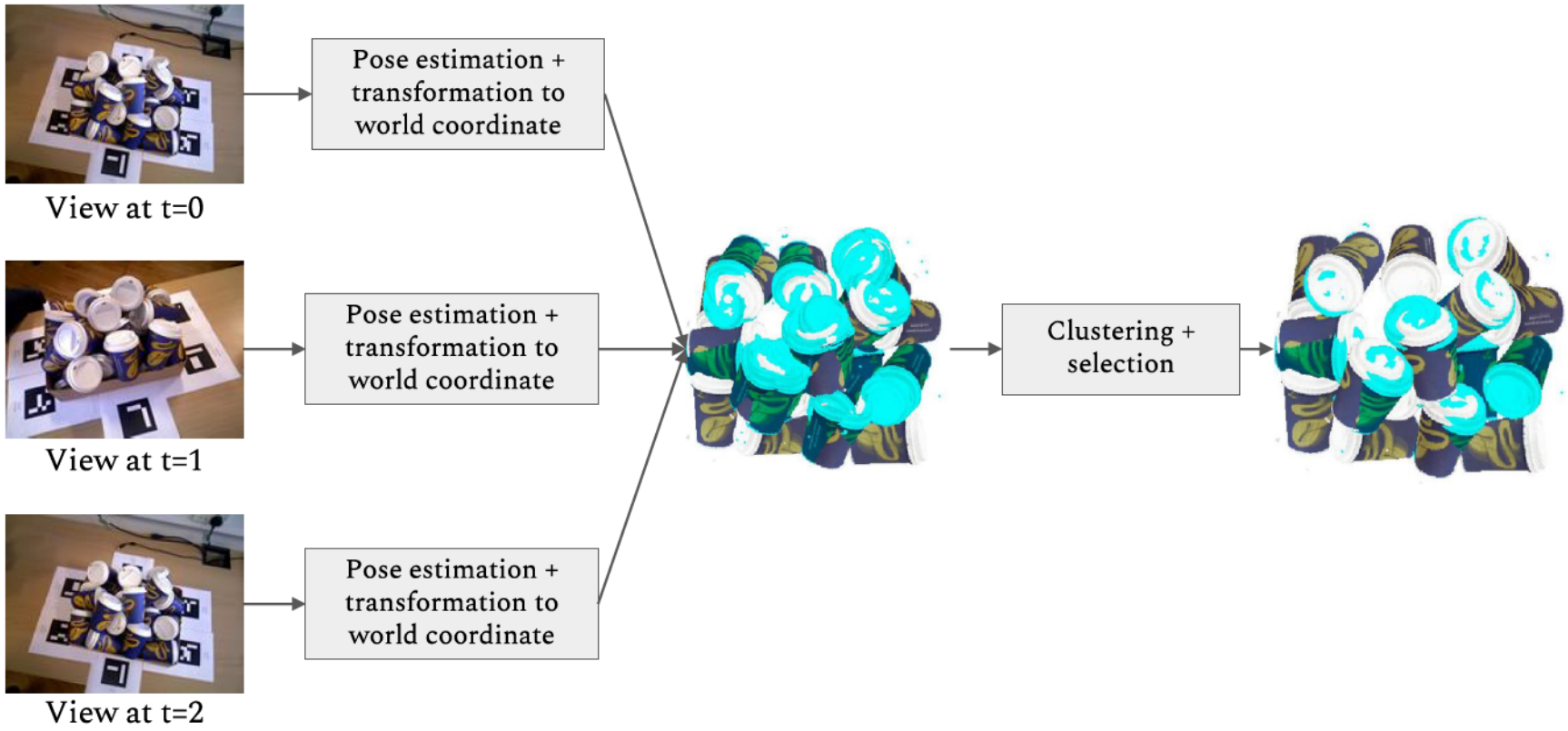


Figure 4.2: Multi-view hypothesis fusion. Since the hypothesis are estimated in camera coordinate, they need to be transformed into a world coordinate using the camera pose. Hypotheses are clustered based on the distance between the centre of the hypotheses. For each cluster, the hypothesis with the highest accuracy among all the hypotheses is chosen to represent the cluster.

depth map that lie inside the mask are considered as inliers. The local fitting between p and q is measured by $\delta(p, q)$ and is defined as follows:

$$\delta(p,q) = \begin{cases} \frac{1}{2}(1 - \frac{\|p-q\|_2}{\epsilon}) + \frac{1}{2}(n_p \cdot n_q), & \text{if } \|p-q\|_2 < \epsilon \\ 0, & \text{otherwise} \end{cases}, \quad (4.1)$$

where n_p and n_q denote the normal vectors at p and q respectively. The verification score c is calculated as the mean value of $\delta(p, q)$ for all the inlier points.

Objects feature representation. When encoding object features, 6D object pose hypothesis information is not directly used given that some estimated object poses are of poor quality to provide useful enough information. Instead, I propose to use a more reliable and simpler feature representation to encode object hypotheses. The output of the hypotheses selector is a set of object features $\mathcal{B}_t$ for k object hypotheses. Each element in $\mathcal{B}_t$ consists of a tuple (b^i, d^i, c^i), where b^i is the normalised 4-dimensional object 2D bounding box coordinates, d^i is the distance of the hypothesis from the viewpoint in depth values on the camera coordinates and c^i is the verification score of the hypothesis. Bounding box coordinates are directly used for b^i, and d^i is approximated by averaging the mean value of depth map values corresponding to the segmentation mask of the 6D object pose estimator. The value from the object hypothesis in depth axis is not directly used as it is often incorrect, notably when the object is highly occluded.

4.3.3 *Object attention module*

At a given time step t, the agent observes a number of object hypotheses and makes a choice on which next viewpoint to

Algorithm 1 Attention module

Data: $\mathcal{B}_t$ features of k objects at time step t
Result: o_t object feature and object index m_t for the policy network
for $b^i \in \mathcal{B}_t$ **do**
  $g^i := FC(b^i, d^i, c^i)$ `// fully connected network`
$g^{global} := meanpool(g^1, ...g^k)$
for $b^i \in \mathcal{B}_t$ **do**
  $w^i := Selector(concat(g^i, g^{global}))$ `// object score`
$\bar{w}^1, ...\bar{w}^k = softmax(w^1, ...w^k)$
$m_t = argmax(\bar{w}^1, ...\bar{w}^k)$ `// attended object index`
$o_t = concat(g^m * \bar{w}^m, b^m, d^m, c^m)$ `// individual object feature`

visit at $t+1$. To guide the agent on that decision, an attention mechanism is proposed that makes the agent focus on the most uncertain object and move accordingly. The attention mechanism receives as input the set of object features $\mathcal{B}_t$ from the previous module and returns an individual object o_t feature of the object to reason about and its index m_t in a similar way as Devin et al. (2018) and Wang, Devin, Cai, Yu and Darrell (2019). First, a representation of each object in $\mathcal{B}_t$ is extracted using a fully connected network. These individual features are aggregated using a mean-pool layer leading to a global object representation that is concatenated to the individual feature. A selector network takes these concatenated features and outputs a scalar score indicating the importance of the object, which is normalised using a softmax layer. The object with the highest score is then selected, and its features are given to the policy network. Note that gradients flow from the policy network, guiding the attention module to the object of interest in the active setting. A description of the algorithm can be found in Algorithm. 1. *Selector*$(\cdot)$ is a linear mapping from object features to attention score which is optimized jointly with the policy.

4.3.4 *Policy learning*

Formulation of reinforcement learning problem. The decision process of choosing an action a_t leading to a viewpoint v_{t+1} is formulated within a reinforcement learning (RL) framework. The next viewpoint is strategically chosen to lead to an optimal set of hypotheses $\hat{x}^*$ that best describes the scene. The optimization problem is defined as the maximization of a parametric function $J(\theta)$ defined as $J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\theta)}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$, where $p_\theta(\theta)$ is the distribution over all possible trajectories following the parametric policy π_θ. The term $\sum_{t=0}^{T} \gamma^t r_t$ represents the total return of a trajectory for a horizon of T time steps and a discount factor $\gamma \in [0, 1]$.

Reward function. The reward function involves three different terms: (i) individual pose estimation accuracy; (ii) change in distance from the camera to the attended object and (iii) penalization for long-distance camera movements. The first term is defined as follows:

$$r_t^{e_{ADD}} = e_{ADD}(\hat{x}_{t+1}^{m_t}, x_{t+1}^{m_t}) - e_{ADD}(\hat{x}_t^{m_t}, x_t^{m_t}), \tag{4.2}$$

where index m_t denotes the object index selected by attention network described in sec.4.3.3 at time t. e_{ADD} is the most widely used 6D object pose error function in the literature and is the average Euclidean distance of model points proposed by Hinterstoisser et al. (2012). It is defined as the average Euclidean distance of the estimated pose $\hat{x}$ with respect to the ground-truth pose x of an object model.

I hypothesize that a viewpoint closer to the object of interest is more likely to give better pose estimation. It can be used to guide

and accelerate the policy learning and this leads to the second term defined as:

$$r_t^{dist} = \|\hat{x}_{t+1}^m[xyz] - v_{t+1}[xyz]\|_2 - \|\hat{x}_t^m[xyz] - v_t[xyz]\|_2, \quad (4.3)$$

where $[xyz]$ is the translation vector from a 6D pose. The full reward function is the combination of the two above, and another term penalizing long-distance camera motions aiming to minimize the total distance travelled d:

$$\begin{aligned} r_t = (1-\alpha) r_t^{eADD} + \alpha\beta r_t^{dist} - \\ (1-\alpha)(1-\beta)(\|v_{t+1}[xyz] - v_t[xyz]\|_1). \end{aligned} \quad (4.4)$$

where α and β are hyperparameters in $[0, 1]$ weighting the influence of the different reward terms.

State space. The state vector s_t in the framework consists of three concatenated components. First, the object feature o_t that comes from the attention module presented in Algorithm 1. Second, the current camera pose position $v_t[xyz]$. Last, a history vector H_t that encodes previous camera positions for T steps, which is initialised to zeros and it is filled at each time step.

Action space. The action space is defined as a two-dimensional continuous space spanning azimuth and elevation angles the camera viewpoint of a hemisphere centred in the stack of objects. More specifically, action at time step t is defined as $a_t = \{\phi_t^a, \phi_t^e\}$ where ϕ^a and ϕ^e represent azimuth and elevation angles of the camera respectively. Given the finite nature of the viewpoint space, the angles from a_t are mapped to the closest viewpoint in $\mathcal{V}$.

Policy network The policy function π_θ is represented as a DNN with parameters θ that outputs the mean and standard deviation of a Gaussian distribution. To optimize θ , policy gradients method (Schulman et al., 2017) is used. These methods opti-

mize $J(\theta)$, where the gradient of the expected return $\nabla_\theta J(\theta)$ is estimated with trajectories sampled by following the policy.

4.4 EXPERIMENTAL EVALUATION

4.4.1 *Dataset and experimental setup*

For synthetic experiments, Bin-picking dataset with two different object models are used for training and testing: Coffee Cup from Doumanoglou et al. (2016) dataset and Bunny model from Brégier et al. (2017) dataset. For each model, 40 scenes are generated with a different number of objects and random pose configurations. Objects with low visibility not detected in any viewpoints are removed from the ground-truth and are not included in the evaluation. For each object, 30 scenarios are used for training and ten scenarios are used for testing. Each scenario has 100 views on the upper hemisphere with a radius of 80 cm. The view grid consists of 5 elevation, and 20 azimuth levels and both RGB and depth images are rendered with OpenGL. The number of instances varies randomly between 15-20 for Coffee Cup and 7-12 for Bunny due to the size and shape difference. For e_{ADD}, all detectable objects are taken into account, and the undetected object are assigned the error of 50mm, which is more than 3 times the acceptable threshold.

For the real data experiment, coffee cup scenario from Doumanoglou et al. (2016) is used. Since the viewpoints of the dataset are not evenly distributed, a view grid is constructed in the same ways as the synthetic environment, and the image from viewpoint closest to the grid viewpoint is used as the observation. During training and testing, 2D bounding box coordinate is rotated around the depth axis to correct the in-plane rotation. The inference time for the pose estimator depends on the number of objects in the scene.

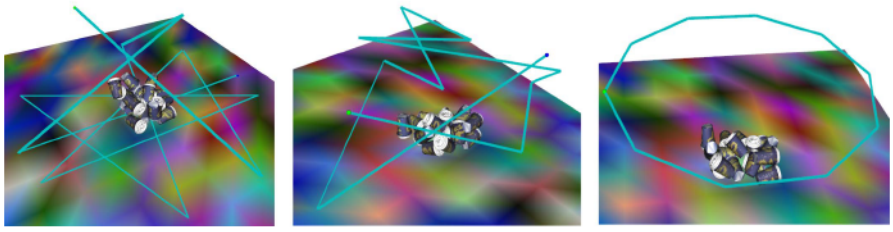


Figure 4.3: Baseline policies. (Left) Sample image of Maximum Distance policy baseline (Middle) Sample image of Random policy baseline (Right) Sample image of Unidirectional policy baseline

In the case of 14 objects, the pose estimator outputs hypotheses at 2 Hz. The rest of the pipeline, including object accumulation, clustering and policy network, operates at 25 Hz. It is possible to design a pipeline to leverage intermediate image data acquired while travelling from v_t to v_{t+1}, however, it is less practical due to the processing time.
The framework implementation is built with PyTorch and PPO(Schulman et al., 2017) implementation from Kostrikov (2018).

4.4.2 *Baseline policies*

The proposed active pose estimation system is tested against a variety of baselines. The first baseline **"Random"** samples random actions from a fixed Gaussian distribution. The second baseline **"Unidirectional"** moves the camera in one horizontal direction around the scene. The trajectory is generated such that it completes one full revolution around the scene. Elevation of the viewpoint is fixed to be 45 degrees which provides a balanced view between occlusion and variability in each view. The third baseline, **"Maximum distance"**, selects viewpoints with the longest distance from previous viewpoints. This baseline test whether the most informative views are simply far-apart views. Sample trajectories of baselines are visualised in Figure. 4.3.

Table 4.1: Quantitative results. Detection rate, e_{ADD} and distance travelled evaluated with the different baselines on synthetic environment. Maximum episode length set to 5. All experiments are repeated 10 times with different starting viewpoint and averaged.

Policy	Coffee cup			Bunny		
	Distance d ↓	e_{ADD} (mm) ↓	Detection rate ↑	Distance d ↓	e_{ADD} (mm) ↓	Detection rate ↑
Random	4.97	13.48	0.76	3.97	26.69	0.19
Maximum Distance	7.66	14.02	0.75	7.68	26.83	**0.46**
Unidirectional	**2.88**	14.32	0.74	**2.88**	27.12	0.44
Proposed	3.71	**11.12**	**0.80**	**2.43**	**25.35**	**0.46**

4.4.3 *Evaluation metrics*

Pose accuracy for all objects are evaluated with e_{ADD} scores. For symmetric objects such as coffee cup, the hypothesis prediction corresponding to the axis of symmetry are ignored when e_{ADD} was calculated. Following the standard practice (Hinterstoisser et al., 2012), objects with e_{ADD} error less than 10% of the object diameter is considered to be a correct hypothesis. All pose errors are measured in mm. The detection rate is defined as the ratio of number of objects with correctly estimated pose to the total number of objects in the bin in each scenario. Distance travelled is the total distance the camera moved where it is measured along the surface of the hemisphere as it is unrealistic for the sensor to move through the objects.

4.4.4 *Synthetic dataset*

Table. 4.1 shows the mean pose error and the correct detection rate on the synthetic dataset. To obtain the results, episode lengths T are held constant at five steps. For Coffee Cup object, the proposed approach consistently outperforms all baselines in both e_{ADD} and detection rate metric by a significant margin. In most

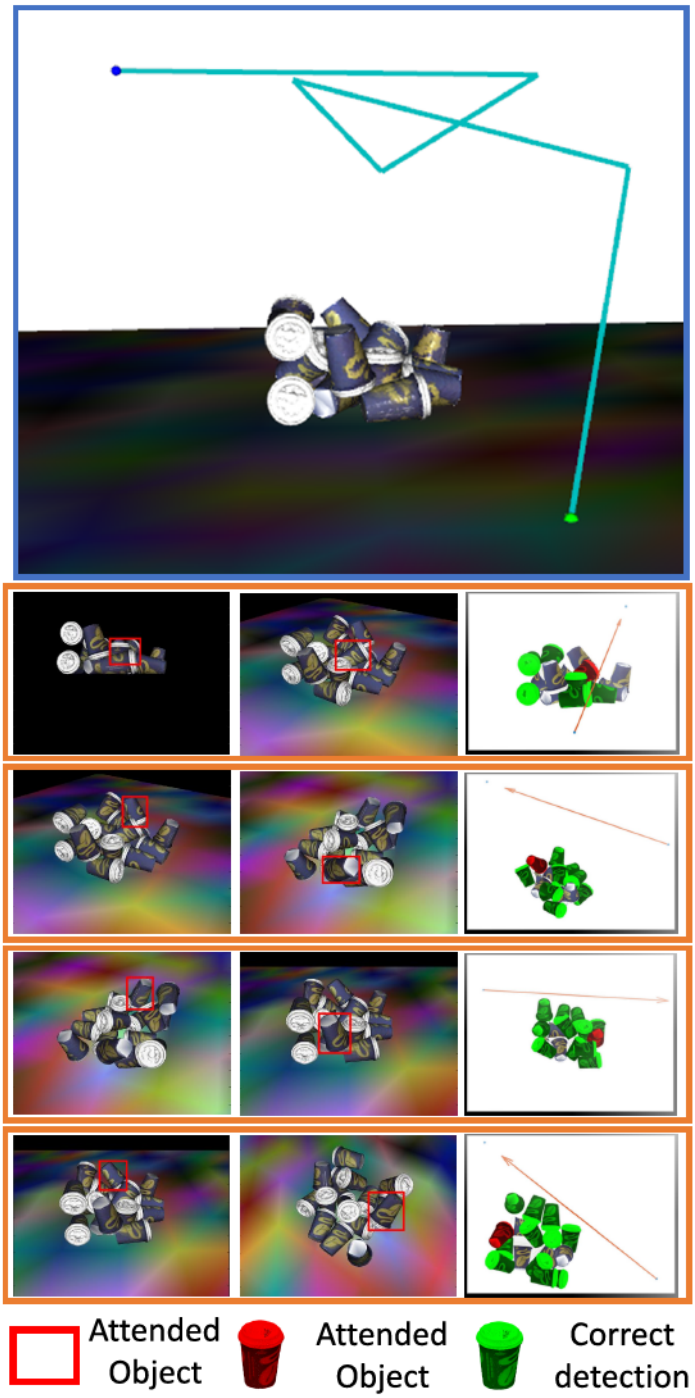


Figure 4.4: Visualisation of attended objects during a trajectory. (Left) An example of the learned trajectory for the coffee cup dataset. (Right) Every column represents one time step (Top and middle row) Image from the viewpoint before and after taking the action. (Bottom row) 3D visualization of the action for each time step.

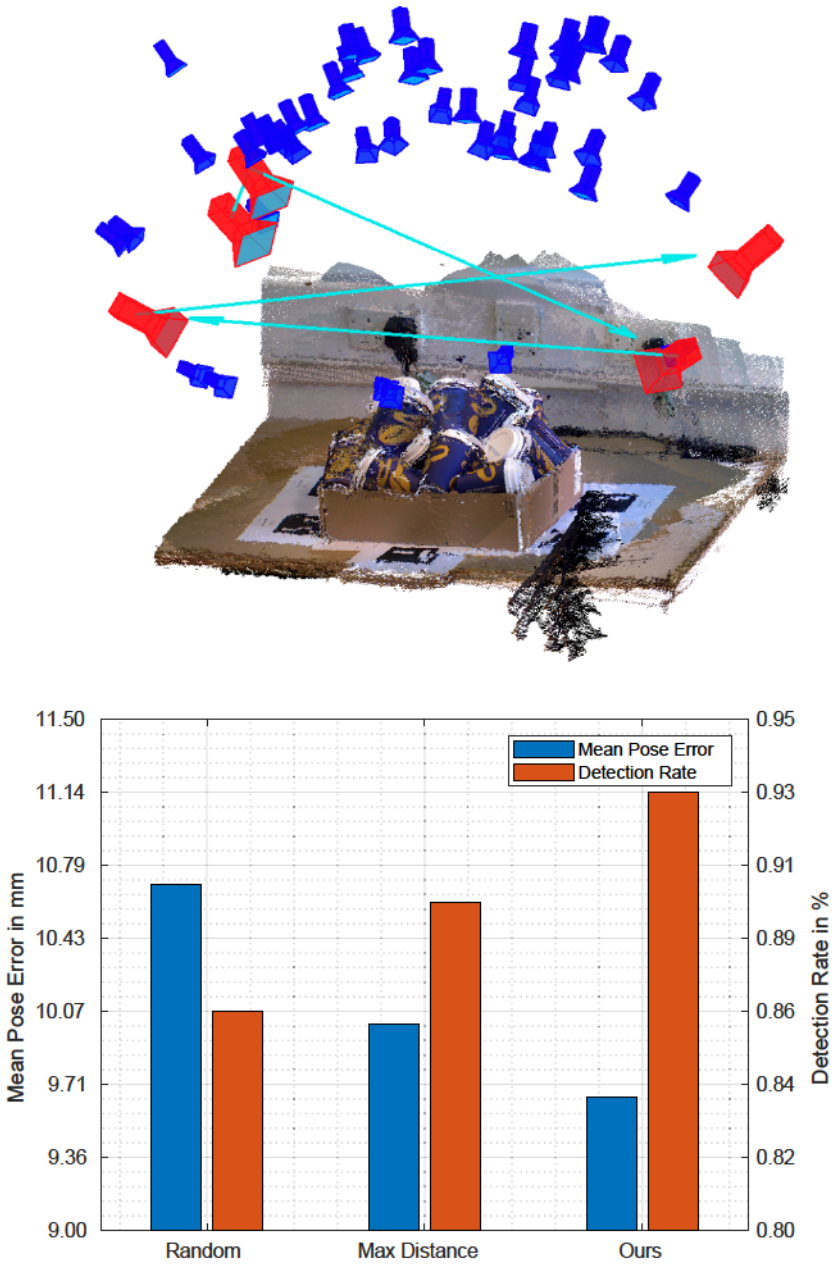


Figure 4.5: Evaluation on real dataset. (Top) Visualization of a sample trajectory. Blue camera represents a set of allowable viewpoints and red cameras show the viewpoints selected by the proposed system. The arrow shows the order of view selection. (Bottom) Performance of different baselines on the real dataset.

cases, *Unidirectional* policy results in low distance d as the agent only moves horizontally. For Bunny object, the proposed method outperforms all baselines in all metrics with a narrower margin due to the smaller number of instances in the scene. Figure.4.4 shows how the agent behaves at each time step of the trajectory. The agent selects the object with the lowest confidence, which is expressed as the verification score, but also not too far away to reduce the distance travelled. The attention shifts to different objects in different time step, and it can be noted the number of the correctly detected object (highlighted in green) increases every step.

4.4.5 *Real world dataset*

Figure. 4.5 shows a sample trajectory generated by the learned policy network and quantitative comparison with other baselines. Unidirectional baseline is not included since the distribution of viewpoints in the real dataset is not in a grid form or uniform. Since most of the viewpoints are densely populated on the top part of the scene, the learned policy tends to select views with lower elevation resulting in a behaviour similar to maximum distance baselines.

4.4.6 *Ablation experiments*

α **and** T **parameters.** Experiment with different trajectory is presented to show how the performance changes within the spectrum between single-shot pose estimation ($T = 0$) and all-view pose estimation ($T = 99$ in this case). Figure. 4.6 shows the performance indeed increases with more views, but the benefit saturates after $T = 10$. For α, excessively emphasizing the r^{dist} term results in the agent to give attention to the furthest object

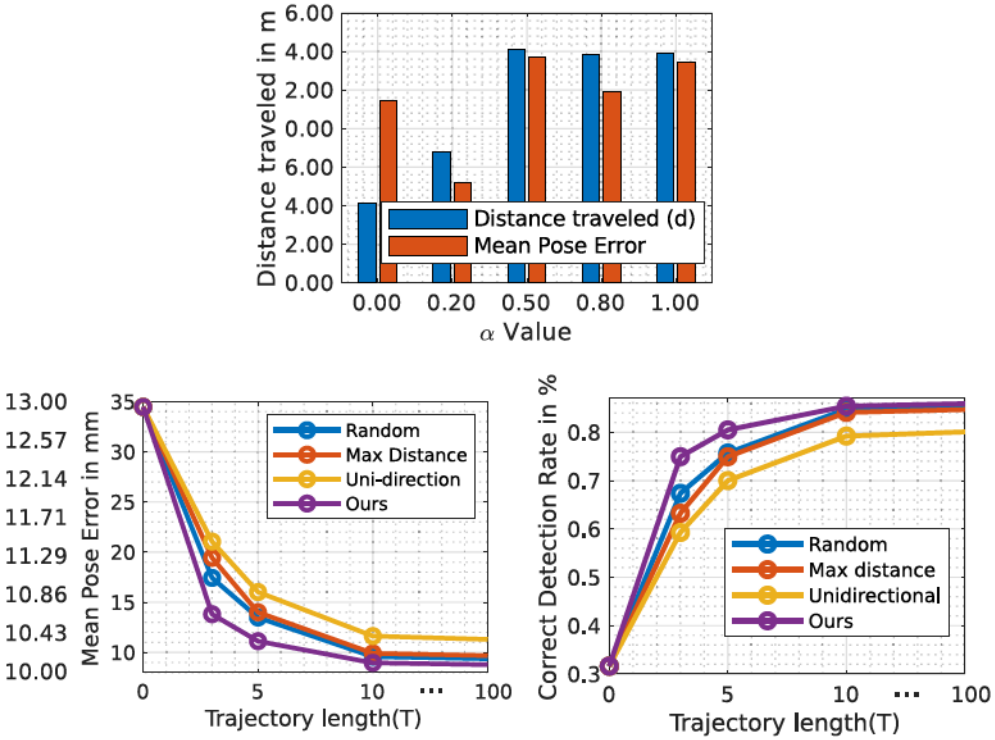


Figure 4.6: Performances with different hyperparameters. (Up) Distance travelled and mean pose error with respect to α values. Change in mean pose error (Left) and detection rate (Right) with different number of maximum episode length (T).

and move the camera towards the object to maximize the reward, increasing the travel distance. Whereas it is difficult for the agent to learn a policy without r^{dist} term since during training often there are multiple views around the object of interest which are equally favourable.

Attention module. To verify if the attention module is properly learning to automatically select an object, a baseline which assigns attention to the object with the lowest verification score is tested. Compared to the verification score-based selection, the attention baseline improved the detection rate by 3% and mean pose error by 9%.

Qualitative evaluations. More qualitative evaluation results, including the attention histogram for detected object for better

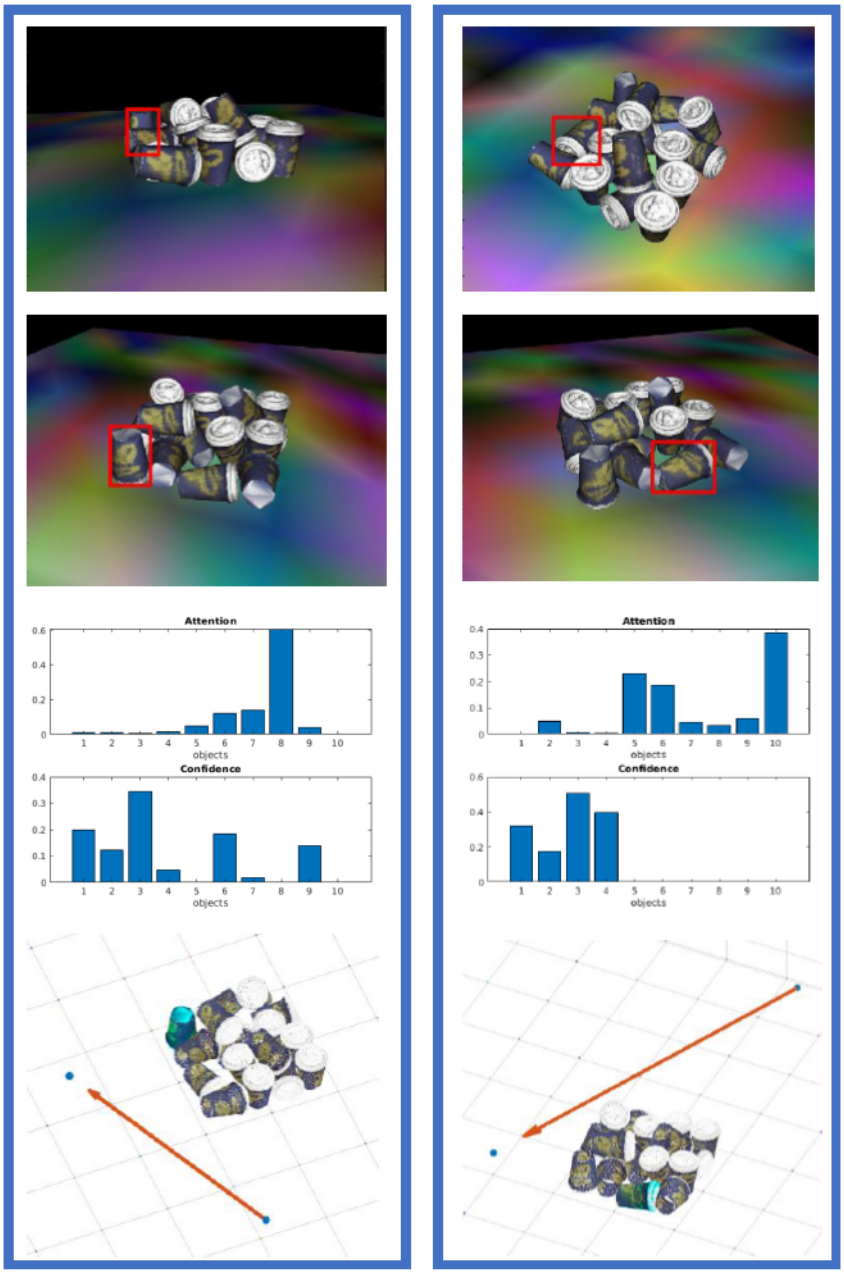


Figure 4.7: Relationship between attention, verification score and action. (First row) Images of the scene before taking action. Attented object is highlighted in red box. (Second row) Images of the scene after taking action. (Third row) Bar graph showing the object score($\bar{w}^1$ in the manuscript) and verification score(shown as confidence). (Fourth row) 3D visualization of camera trajectory.

insights, are shown in Figure. 4.7. It can be noted the selector network tends to choose the object with the most uncertainty.

4.5 SUMMARY

A framework to deal with the active 6D object pose estimation problem in cluttered scenarios is presented in this chapter. The framework is formulated within a reinforcement learning and carefully designed all the different components achieving superior performance compared to different baselines. However, some part of the framework can be improved for the better efficiency and performance. Although any off-the-shelf pose estimator can be used in the proposed system, it is wasteful to estimate poses of all the objects in the scene from scratch. Integrating a pose estimator which utilises the estimated poses from previous time step as a prior to the system could potentially increase the performance. In terms of experiments, I believe there is a margin of improvement in terms of performance and real-world framework evaluation given the lack of proper datasets.

5

SELF-SUPERVISED LEARNING FOR OBJECT POSE ESTIMATION

5.1 OVERVIEW

A recent trend in 6D object pose estimation is to use DNNs, which have shown strong performance when training data is available in large scales (Park et al., 2019; Rad and Lepetit, 2017). DNN-based methods can be trained using rendered images of 3D object models, as a large quantity of training images can be synthesised where pose labels are automatically generated. However, as it will be shown in the experiments in this chapter, naively training on synthetic images generalises poorly to the real domain (Sundermeyer et al., 2018; Zakharov, Shugurov and Ilic, 2019). Although visually similar, rendered images and real images are different in many ways due to multiple factors from compression effects to lens vignetting. Therefore DNNs often exploit real images with accurate 6D pose labels (Li et al., 2019; Park et al., 2019; Rad and Lepetit, 2017; Tekin et al., 2018) in addition to the synthetic data. Unfortunately, it is cumbersome to obtain real data with high quality 6D pose annotations in large quantities. To overcome this limitation, supervised and unsupervised domain adaptation methods have been proposed (Bousmalis et al., 2017; Ganin et al., 2016; Rad et al., 2018) to learn from both real and synthetic domain.

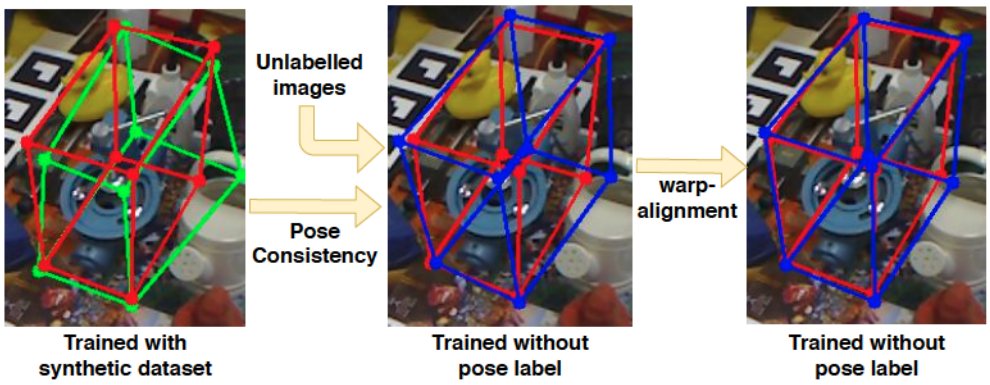


Figure 5.1: Illustration of the proposed framework. A self-supervised approach is proposed in this chapter that leverages a pose consistency loss and a warp-align framework to learn a robust 6D object pose estimator without using real pose labels. Red box shows the ground-truth pose and green and blue boxes represent estimated pose from model trained with synthetic data only and the proposed self-supervision respectively.

In this chapter, I propose two frameworks: self-supervised learning for domain adaptation and unsupervised warp-align framework for fine-tuning. The self-supervised method aims to achieve a domain-invariant 6D object pose estimator via pose consistency without using pose labels in the real training data. Inspired by the method of Zhu et al. (2017), which learns image-to-image translation with cycle consistency, I propose a framework based on pose consistency which guides the model to learn to produce consistent pose estimations for both real and synthetic images. This is achieved by generating image-label pairs in the synthetic domain conditioned on the pose predictions in the real domain using a differentiable renderer during the training. The method works with colour images only, as opposed to previous self-supervised methods which require extra modality such as depth map for fitting (Wan et al., 2019; Wang et al., 2020). It is desirable as capturing depth information requires special hardware which is not always available, particularly in outdoor applications. Moreover, the framework is flexible and virtually any object

pose estimator could benefit from it. In this chapter, the proposed framework is tested with two state-of-the-art methods, namely BB8 (Rad and Lepetit, 2017) and Pix2Pose (Park et al., 2019).

Additionally, the trained estimator with the proposed self-supervised can be further fine-tuned by exploiting the relation between the camera poses and object poses. It is named warp-align framework in this thesis. Inspired by approaches in monocular depth estimation literature (Godard et al., 2019; Ranjan et al., 2019), the observation is that the camera poses can be inferred from the predicted object poses to warp the input images into another view to be perceptually compared and provide supervisory signal to fine-tune the pose estimator. The overview of the proposed frameworks are illustrated in Figure. 5.1. Unlike the method in Chapter 4 where the scene needed to be static, the training dataset can be generated from a video in the dynamic environment where the object in the scene is moving.

5.2 RELATED WORK

5.2.1 *Datasets*

Annotating geometric labels such as 6D object pose is costly, often involving special hardware (Ionescu et al., 2014; Yuan et al., 2017) and only captures limited views as the space of possible view-points is immense. Yuan et al. (2017) uses a magnetic tracking system with six magnetic 6D sensors to automatically annotate the hand joint positions. Other dataset often use a set of carefully calibrated cameras and annotate pose of an object by iteratively minimising the reprojection errors. Efforts have been made to create a real RGB dataset with high-quality labels such as Pix3D (Sun et al., 2018), yet it is limited due to the time and

effort required.

5.2.2 *Real-synthetic domain gap*

Some existing methods render synthetic images from a 3D model with perfect labels to train the network in the synthetic domain and deploy them in real-world (Kehl et al., 2017). However, training purely by synthetic images does not generalise well to real images (Zakharov, Shugurov and Ilic, 2019). Although visually similar, the rendered and real images are different in many ways due to factors such as compression effects or lens vignetting. To overcome this shortcoming, mainly two different approaches have been proposed for 6D object pose estimation: domain randomisation (Sundermeyer et al., 2018; Zakharov, Kehl and Ilic, 2019) and domain adaptation (Bousmalis et al., 2017). Domain randomisation hypothesises that enough variability in augmentation for simulated images will generalise to real images (Tobin et al., 2017). However, the domain randomisation is limited to the type of parameters (*e.g.* brightness, contrast, etc.) being randomised, selected heuristically. The manually chosen parameters may not be the source of the domain gap causing the performance degradation. Recently, Zakharov, Kehl and Ilic (2019) have proposed a method that learns the optimal weights for augmentation modules which helps achieve maximum domain confusion. Domain adaptation is categorised into unsupervised (Bousmalis et al., 2017; Ganin et al., 2016) and supervised (Rad et al., 2018) learning. Supervised domain adaptation uses image-label pairs from both the source and target domain and directly learns the mapping between the two representations (Georgakis et al., 2018; Rad et al., 2018). These methods show promising results but require labels in the target domain. Recently, unsupervised domain adaptation with Generative Adversarial Networks (GANs) (Bous-

malis et al., 2017; Hoffman et al., 2018; Murez et al., 2018) has been proposed to generate target domain images without labels. Another example of an unsupervised method is GRL (Ganin et al., 2016), where domain-invariant features are developed to deceive domain classifiers. However, often the performance of these methods is suboptimal as the methods learn to match the distributions of the domains without considering the task at hand (Lee et al., 2019). Also, such methods are prone to overfitting and degeneration of performance is observed if the samples are out of distribution (Sundermeyer et al., 2018; Zakharov, Kehl and Ilic, 2019).

5.2.3 *Self-supervised pose estimation*

Recently, self-supervised methods have shown that a model can be trained to predict pose by constraining the prediction to be consistent with input data by fitting (Mees et al., 2019; Palazzi et al., 2018; Wan et al., 2019; Wang et al., 2020). Although they do not require real data with pose labels, they use an alternative form of ground truth as a weak supervisory signal. Wang et al. (2020) have used visual and depth alignment to learn 6D pose estimation without the label. However, in the experiment, it has been shown the accuracy largely relies on the geometric information from the depth map. Supervision from visual alignment does not provide reliable supervision as when the real image and the rendered image of the predicted pose are aligned, the domain gap between them makes the supervision less meaningful. Wan et al. (2019) have exploited both depth maps and multi-view data to achieve high performance. Self-supervised learning has also been used to learn meaningful feature representations by applying simple geometric augmentation to input images such as rotation (Chen et al., 2019; Tran et al., 2019). In this chapter, I

propose to apply the silhouette of the predicted pose to augment the input image. This technique is shown to be crucial for bridging the synthetic—real domain gap in the experiments.
Another domain where self-supervision is widely used is monocular depth estimation (Godard et al., 2019; Ranjan et al., 2019; Zou et al., 2018). The idea is to estimate both camera viewpoint and depth map for two consecutive frames, which enables to warp source frame to the target frame. A photometric loss between the warped frame and the target frame can serve as an unsupervised proxy loss. Inspired from this idea, I propose an unsupervised learning framework which further enhances the pose estimator trained with the self-supervision. The core observation is that the relative viewpoint between two images can be calculated from the estimated poses. Also, rendering depth map using the object model with estimated pose via differentiable renderer can provide geometric data needed without explicitly predicting pixel-wise depth values.

5.3 PROPOSED METHOD

In this chapter, I introduce two frameworks: self-supervised learning for domain adaptation and unsupervised warp-align framework for fine-tuning.

The self-supervised method results in a domain-invariant 6D pose estimator via *pose consistency* without pose labels on real RGB images. I assume that the primary differences between the two domains, real and synthetic, are in low-level such as illumination and noise rather than high-level geometric variations (Bousmalis et al., 2017; Kulkarni et al., 2019). The proposed method utilises the silhouette of the object which requires the geometric shape between the 3D model and the object in the scene to be identical. The proposed method adopts learning-by-synthesis in

an end-to-end manner, via differentiable PnP and a differentiable renderer. The primary advantage of the proposed framework over the prior arts is that the proposed method gets supervision from the discrepancy between the estimated poses on different domains rather than getting a training signal from fitting. This eliminates the need for depth maps (Wan et al., 2019; Wang et al., 2020) which may not be available in common applications.

In the second part of the chapter, I present a warp-align step that further enhances the accuracy of the pose estimator, again without requiring the real pose label.

5.3.1 *Self-supervised 6D Object Pose Learning*

Let $\{\mathcal{I}^g, y^g, \mathbf{h}^g\}$ denote a synthetic generated dataset and $\{\mathcal{I}^r\}$ a real one. $\mathcal{I}$ and y represent images and object poses (3D rotation and translation), respectively. $\mathbf{h}$ is an intermediate representation, which is the output of a 6D object pose estimator with parameters θ and denoted $\Phi(\mathcal{I};\theta)$, and its representation varies depending on the used baseline. For instance, $\mathbf{h}$ are the projected pixel locations of a 3D bounding box for the baseline BB8 (Rad and Lepetit, 2017) and the normalised object coordinates and object mask for Pix2Pose (Park et al., 2019). It can be computed as $\mathbf{h}^g = \pi(y^g, \mathcal{M}, \mathbf{K})$, where $\pi(\cdot)$ is a function that maps the object pose to the intermediate representation for a given object model $\mathcal{M}$, and the camera intrinsic parameters $\mathbf{K}$. During training (see Fig.5.2), the baseline pose estimator infers the intermediate representation from a real RGB image, $\mathbf{h}_1 = \Phi(\mathcal{I}^r;\theta)$. $\mathbf{h}_1$ is used to yield $\mathcal{R}(\text{PnP}(\mathbf{h}_1, \mathbf{K}), \mathcal{M}, \mathbf{K}) = \{r_{\mathbf{h}_1}, m_{\mathbf{h}_1}, y^g_{\mathbf{h}_1}\}$ via the PnP algorithm $\text{PnP}(\cdot)$ and the renderer $\mathcal{R}(\cdot)$. The functional outputs are the new synthetic image $r_{\mathbf{h}_1}$, its silhouette $m_{\mathbf{h}_1}$ and the estimated object pose $y^g_{\mathbf{h}_1}$.

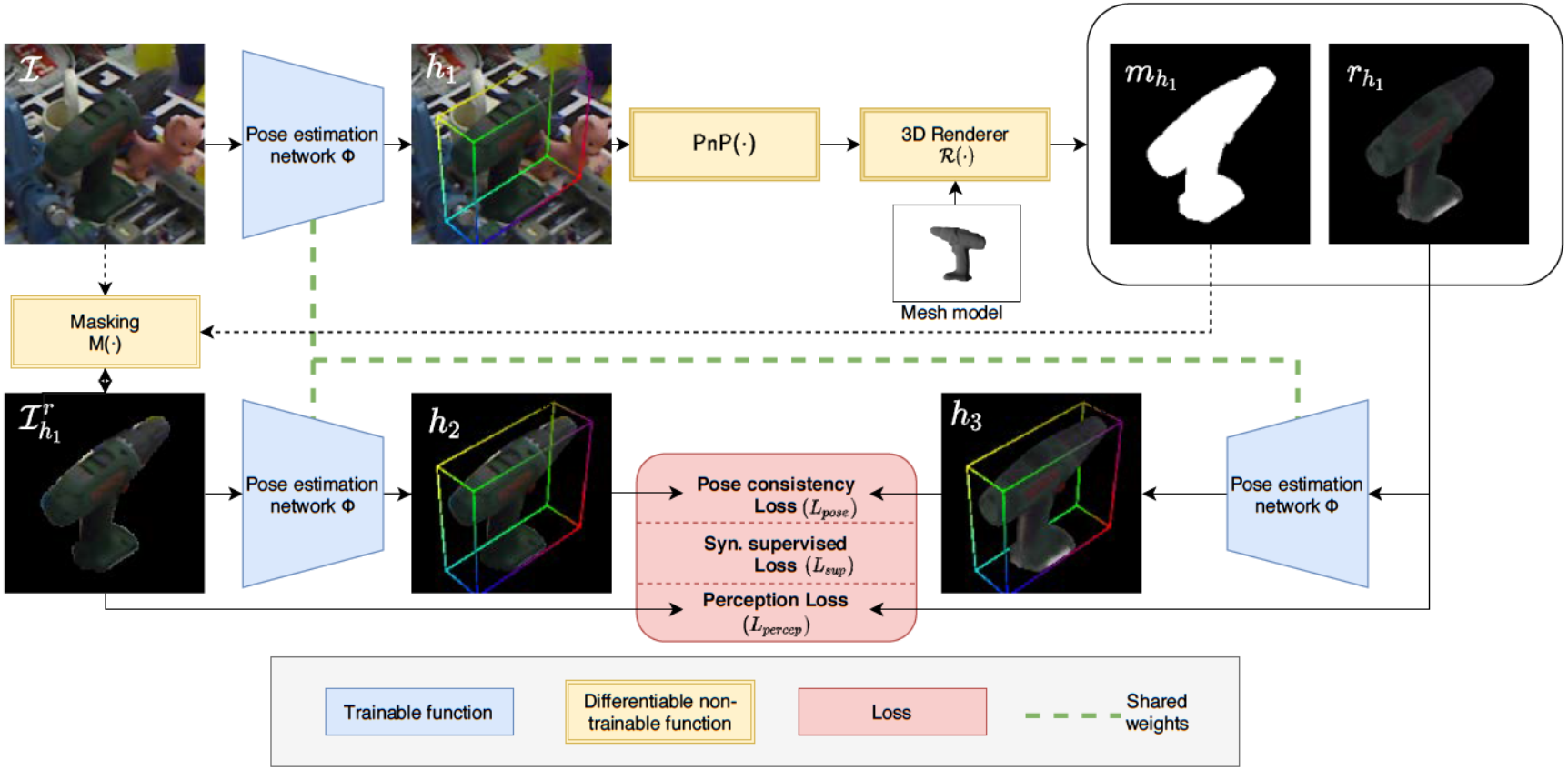


Figure 5.2: Proposed self-supervised framework. The figure shows the proposed self-supervised framework using pose consistency. I start training from a baseline pose estimator trained purely with synthetic RGB data. Given real unlabelled data, the pose is estimated and an image is rendered through differentiable renderer based on the prediction. Object poses of input images augmented with the estimated silhouettes and rendered images are enforced to be consistent.

A masking function $M(\mathcal{I}^r, r_{\mathbf{h}_1}) = \mathcal{I}^r_{\mathbf{h}_1}$ outputs the masked real image using the silhouette. Finally, the pose estimator predicts the intermediate representation for both the masked image $\mathcal{I}^r_{\mathbf{h}_1}$ and rendered synthetic image $r_{\mathbf{h}_1}$, that is $\Phi(\mathcal{I}^r_{\mathbf{h}_1}; \theta) \rightarrow \mathbf{h}_2$ and $\Phi(r_{\mathbf{h}_1}; \theta) \rightarrow \mathbf{h}_3$. As shown above, object poses are estimated multiple times across different domains, but the weights θ are shared. In the following, I present each component in detail:

Silhouette masking M**.** Its function is to transform the real input image by removing the background with the silhouette rendered using the estimation $\mathbf{h}_1$: $M(\mathcal{I}^r, m_{\mathbf{h}_1}) = \mathcal{I}^r_{\mathbf{h}_1}$, where M is the masking function to remove the background given a silhouette. The masked image is used to estimate the pose $\mathbf{h}_2$, which is encouraged to be consistent with $\mathbf{h}_3$. The intuition is that in order to achieve this goal, the estimated pose $\mathbf{h}_1$ needs to be correct, otherwise, the foreground image would be removed resulting in the pose estimator not being able to estimate $\mathbf{h}_2$ consistently.

Differentiable PnP and Differntiable renderer $\mathcal{R}$**.** In order to make the framework end-to-end trainable, a differentiable components are needed to convert the output of the pose estimator to the 6D object pose. I use the implementation from Chen et al. (2020) for differentiable PnP and a differentiable renderer (Kato et al., 2018; Ravi et al., 2020), denoted as $\mathcal{R}(\cdot)$, to generate synthetic images and silhouette given object model, camera parameters and object pose.

Cross domain occlusion augmentation. To make the pose estimator robust to occlusions, randomly sized patches with Gaussian noise are placed on both the rendered image $r_{\mathbf{h}_1}$ and target domain image $\mathcal{I}^r$ in each iteration during training. The part of the silhouette $m_{\mathbf{h}_1}$ occluded by the noise patch is excluded to avoid the masked real image $\mathcal{I}^r_{\mathbf{h}_1}$ containing the noise patch.

Pose consistency loss. The core idea is to encourage the pose estimator to estimate consistent poses for both synthetic and real domains. The pose consistency loss is defined by penalising

the discrepancy between the estimations from different domains. L_{pose} enforces consistency of the transformed vertices of the object model based on the estimated poses, and measure the alignment in 3D space:

$$L_{pose} = \|PnP(\mathbf{h}_2, \mathbf{K})\mathcal{M}_v - PnP(\mathbf{h}_3, \mathbf{K}))\mathcal{M}_v\|_2, \tag{5.1}$$

where $\mathcal{M}_v$ are the vertices position. L_{pose} is agnostic to the baseline pose estimator Φ used, since the output of the pose estimator is converted to homogeneous transformation matrix through differentiable PnP.

Synthetic domain supervised loss. The synthetic domain supervised loss exploits the ground truth pose label that is available in the synthetic domain:

$$\begin{aligned} L_{sup} = \mathbb{E}[\|\Phi(r_{\mathbf{h}_1}; \theta) - \pi(y^g_{\mathbf{h}_1}, \mathcal{M}, \mathbf{K})\|_{huber}] \\ + \mathbb{E}[\|\Phi(\mathcal{I}^g; \theta) - \mathbf{h}^g\|_{huber}], \end{aligned} \tag{5.2}$$

where $r_{\mathbf{h}_1}$ and $\mathcal{I}^g$ are the rendered image via the proposed pipeline and the synthetic images in the training set and $huber$ denotes the Huber loss. Huber loss is less sensitive to outliers than Mean Squared Error (MSE) as it switches between MSE and Mean Absolute Error (MAE) conditioned on the magnitude of the loss. The synthetic domain supervised loss helps to impose constraints on the real data self-supervision, as learning by the pose consistency alone can result in a singularity issue. For instance, the pose estimator can learn to always predict a trivial pose, *e.g.* frontal pose, regardless of the input data, in which the pose consistency loss is minimised. It also helps to learn the prior geometric constraint, *e.g.* the output of the pose estimator always needs to form a valid bounding box in case the Φ is BB8. In general, this loss narrows the gap between synthetic supervision and real self-supervision, preventing the model parameter learning from diverging.

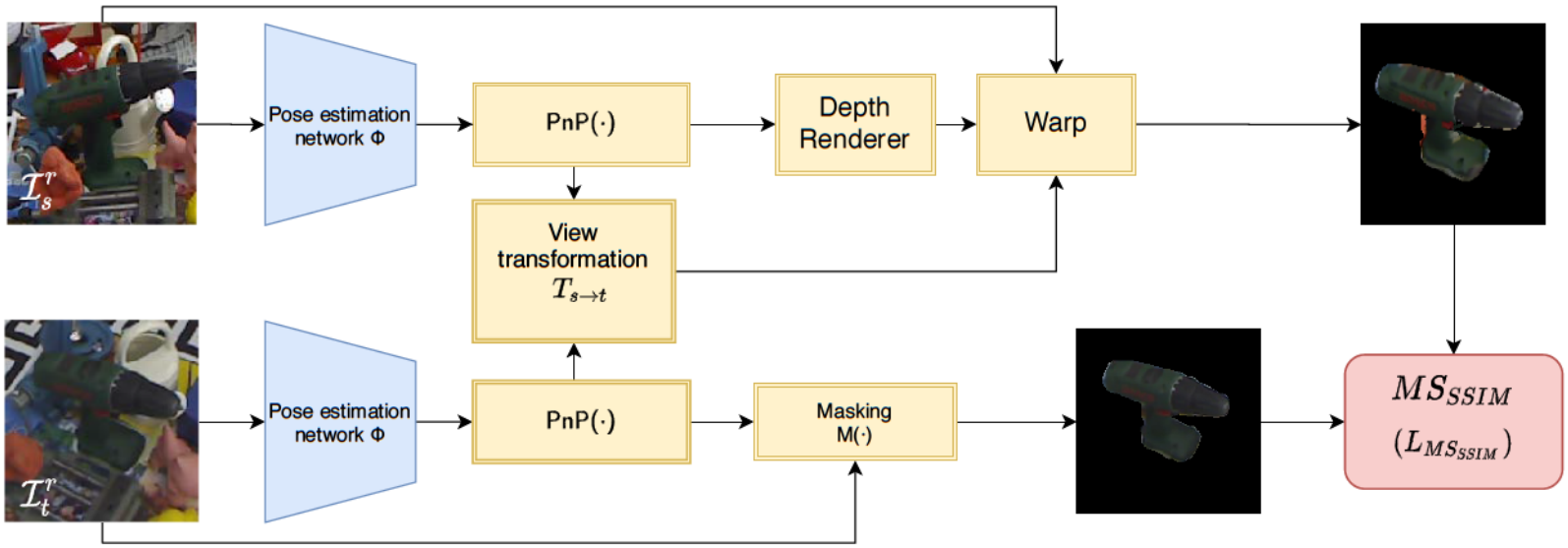


Figure 5.3: Proposed warp-align framework. Using a pair of unlabelled real images, predicted pose is used to estimate the relative camera pose. Together with depth rendered with differentiable renderer based on the estimated pose, source image is warped to the view point of target image. Dissimilarity between the anchor and target image is used to update the pose estimator Φ. The schematic follows the same colour coding as Fig. 5.2.

Overall training objective. In addition to the expected loss over the consistency and the synthetic domain supervised loss, I also use an additional perceptual distance (Zhang et al., 2018) to encourage the rendered image r_{h_1} and the masked real image $\mathcal{I}^r_{h_1}$ to be perceptually similar. I compute the L_2 distance in the feature space of pre-trained VGG (Simonyan and Zisserman, 2015) network between the two images where feature stack from multiple layers are normalised in the channel dimension and denote this loss L_{percep}. The overall training objective is given as:

$$L_{total} = \lambda_{pose} L_{pose} + \lambda_{sup} L_{sup} + \lambda_{percep} L_{percep}, \tag{5.3}$$

where λ_i are constant weights to control the interaction of the losses. For estimator which separately estimate object mask (*e.g.* Pix2Pose), L_1 loss is used for both domains with the ground-truth mask. With the above objective, the proposed framework allows us to learn the pose estimator Φ without the pose labels in the target domain (*i.e.* real RGB images). At the test time, only Φ and PnP are used which is light-weight and fast.

5.3.2 *Warp-Align Fine-Tuning Stage*

Different to the self-supervised learning that learns to estimate poses by leveraging consistency in high-level features such as pose, the goal of the warp-alignment fine-tuning stage is to leverage consistency in lower-level photometric features.

Similar to the self-supervised framework, this step can be used with any off-the-shelf pose estimators without using any pose labels due to its unsupervised nature.

In Fig. 5.3 I show a schematic of the warp-alignment framework. A pair of unlabelled real input images, source image ($\mathcal{I}^r_s$) and target image ($\mathcal{I}^r_t$), are compared and the model learns to minimise the difference between the two images. However, since

the two images are taken from different viewpoints, they cannot be compared directly. The core idea is to transform (warp) $\mathcal{I}_s^r$ to have the same viewpoint as $\mathcal{I}_t^r$, at which the images can be compared directly. In order to achieve the correct transformation and projection to 2D image, two ingredients are required: relative camera view transformation between the two images and depth value for the pixels of the image to be transformed, since the source image needs to be represented in 3D in order to apply 3D camera view transformation.

In 6D object pose estimation problem, the object pose is defined in the camera coordinate space. Hence the camera pose can be defined in the object coordinate space through a simple matrix inversion if the pose is represented as a homogeneous transformation matrix. Since the camera pose for both $\mathcal{I}_s^r$ and $\mathcal{I}_t^r$ can now be defined in the common object coordinate space, the relative view transformation is obtained as follows:

$$T_{s\to t} = PnP(\Phi(\mathcal{I}_t^r;\theta))(PnP(\Phi(\mathcal{I}_s^r;\theta)))^{-1}, \tag{5.4}$$

The depth values of the foreground pixels are required to warp the image to a different viewpoint and that can be obtained by rendering the 3D object model with the pose predicted by Φ. The masking function M is used to remove the background pixels.

3D points of the foreground pixels are then calculated using the rendered depth image and these points are later transformed by $T_{s\to t}$ to be projected onto an image plane.
I use Multi-Scale Structural Similarity Index (MS_{SSIM}) (Zhao et al., 2017) to calculate the similarity between the warped source image $\tilde{\mathcal{I}}_s^r$ and the target image $\mathcal{I}_t^r$ and is defined as follows:

$$L_{MS_{SSIM}} = 1 - MS_{SSIM}(\tilde{\mathcal{I}}_s^r, M(\mathcal{I}_t^r, m_{\mathbf{h}_t})). \tag{5.5}$$

5.4 EXPERIMENTAL EVALUATION

I evaluate the efficacy of the framework through experiments on three datasets, an ablation study is presented to investigate how different components contribute to the performance. I chose two baselines for $\Phi(\cdot)$, namely BB8 (Rad and Lepetit, 2017) and Pix2Pose (Park et al., 2019). These methods are essentially different, which makes them good candidates to demonstrate the generalisability of the proposed framework. BB8 is a holistic pose estimator which regresses pixel locations of the projected 3D bounding box. Pix2Pose uses object coordinates as an intermediate representation and learns to estimate foreground object pixels (mask), object coordinates and confidence scores. In other words, BB8 estimates the pixel location given 3D points of bounding box whereas Pix2Pose estimates a 3D point given a pixel location. Note that own implementations of the baselines are used and both baselines achieve a better performance than the ones reported in the original works. Training the baseline purely on synthetic images serves as a **Lower Bound (LB)** indicator, since the self-supervised training is initialised with the model pretrained with synthetic images. The baseline methods trained with both real and synthetic pose labels serves as an **Upper Bound (UB)**.

I evaluate the approach on the commonly used benchmarks LINEMOD (Hinterstoisser et al., 2012), LINEMOD OCCLUSION (Hinterstoisser et al., 2012), containing 13 and 8 objects respectively, as well as 3 objects from HomebrewedDB dataset following Wang et al. (2020). All reported results are the percentage of correctly predicted poses with AD{D|I} (Hinterstoisser et al., 2012) metric. I set the threshold of 10% of the object's diameter following the BOP standard.

5.4.1 *Implementation Details*

Pytorch (Paszke et al., 2017) is used to train and test the proposed method on a machine with Intel i5 and GTX1080TI. The object models ($\mathcal{M}$) used with the differentiable renderer ($\mathcal{R}$) (Kato et al., 2018; Ravi et al., 2020) were downsampled to reduce the training time if the number of faces is too large.

SELF-SUPERVISION For self-supervised learning, a batch size of 16 is used with the ADAM optimiser (Kingma and Ba, 2014) for both baselines. The initial learning rate is 1e-5 for the first 15 epochs and reduced to 1e-6 for the next 10 epochs. For the experiments presented, I used $\lambda_{\text{pose}} = 0.01$, $\lambda_{\text{sup}} = 1$, $\lambda_{\text{percep}} = 20$ for BB8 and $\lambda_{\text{pose}} = 0.0125$, $\lambda_{\text{sup}} = 3$, $\lambda_{\text{percep}} = 0$ for Pix2Pose. I followed the standard practice to generate training dataset (Peng et al., 2019; Rad and Lepetit, 2017). More specifically, I copy-and-pasted the object image onto random background images. During training, images were cropped 1.3 times the size of the shorter side of the bounding box. The mean position of bounding box was perturbed by Gaussian distribution of the mean value of 0 pixel and a standard deviation of 3 pixels and randomly rescaled. Input images were augmented by: adding Gaussian noises, Contrast normalisation and Gaussian blur. For the LINEMOD dataset, following the standard practice in the community, 15% of the images are used for training but *without* pose labels. The selection of training images follows the strategy in Brachmann et al. (2016). To be robust against inaccurate detection, I randomly perturb the scale and centre position of the bounding box during training. I initialise the network by training with synthetic data only where the early layers are frozen to avoid overfitting to the synthetic dataset. All parameters are updated after initialisation. For all testing, Faster-RCNN (Ren et al., 2015) is used to detect objects and provides accurate bounding boxes with an accuracy

of 99.24% and 97.29% with 50% and 75% IOU respectively, for all objects in the LINEMOD dataset.

WARP-ALIGNMENT Adam optimiser with learning rate of 0.0001 was used where learning rate was droped by 0.1 every 25 epoches. Source images and target images are sampled such that the estimated 3D pose difference between the two is less than 60° as warping into a much different view-point degrades the quality of the warped image. More specifically, 25 samples of real images were sampled from the training dataset, which produced 30300 possible pair combinations between the sampled images. Among them, maximum 25 pairs with the minimum estimated 3D pose difference were selected to form a minibatch of batchsize 25. Warp-align fine-tuning starts training from the pose estimator trained with self-supervised framework.

5.4.2 *Results*

5.4.2.1 *Ablation Study*

Table 5.1 shows the influence of different components for the self-supervised object pose estimation on camera object from the LINEMOD dataset with BB8 as the baseline method. It can be seen that silhouette masking is a critical component of the

Table 5.1: Ablation study on how different terms in the loss and augmentations impact the performance on the camera object for LINEMOD dataset.

Synthetic Supervision	Silhouette masking	Perception Loss	Occlusion	ADD
✓				0.00
✓	✓			29.66
✓	✓			35.31
✓	✓	✓	✓	39.20

Table 5.2: Parameter Analysis: Performance analysis with different weights in the loss for Camera object with BB8. The reported numbers are the recall rate obtained with ADD metric.

λ_{sup} \ λ_{pose}	0.5	1.0
0.5	48.01	46.06
1.0	46.90	48.75

(a)

λ_{sup} \ λ_{pose}	0.5	1.0
0.5	48.10	45.88
1.0	45.23	47.27

(b)

framework. Silhouette masking imposes a strong constraint as the pose consistency loss reduces if the first estimation $\mathbf{h}_1$ is closer to the correct pose, thereby not masking out a portion of the foreground. This intuition is further reinforced in the experiments. Table 5.3 shows that the objects with a more distinctive silhouette for different poses (*e.g.* Benchvise) obtain higher accuracies than those that are not (*e.g.* Duck). Silhouette for a spherical object is invariant to the rotation, which means a wrong 3D pose estimation may still produce the correct silhouette. Perception loss makes training more stable, and occlusion augmentation further boosts the performance since it enforces the estimator to learn more robust feature.

5.4.2.2 *Parameter Analysis*

Table. 5.2 provides changes in performance with different weight parameter values for Camera object in LINEMOD dataset with BB8. Different weights are applied for the pose consistency loss (λ_{pose}), synthetic domain supervised loss (λ_{sup}) and perception loss (λ_{percep}). The results show that although the balanced weights between λ_{pose} and λ_{sup} produce the best performance, the differences are small and the proposed method is robust to different hyperparameters.

Table 5.3: Quantitative results on LINEMOD dataset with ADD metric. (a) Results obtained with the BB8 baseline. (b) Results with the Pix2Pose baseline.

	BB8 Synth only (LB)	BB8+GRL	BB8+PixelDA	BB8+Ours	BB8 with Real Pose Labels (UB)
Ape	0.46	0.74	1.40	27.58	24.79
Bvise	0.75	3.00	0.66	63.07	63.07
Cam	0.00	0.56	1.02	43.37	46.43
Can	0.00	0.96	0.10	51.82	61.61
Cat	0.00	4.52	0.66	47.74	46.39
Drill	0.00	1.26	0.19	50.05	55.86
Duck	0.09	5.98	0.18	34.54	27.11
Eggbox	0.00	29.39	0.00	83.89	88.62
Glue	0.38	8.65	11.85	70.08	84.29
Holep	0.00	2.64	0.91	22.82	34.27
Iron	0.10	12.10	3.60	31.60	76.90
Lamp	0.00	0.93	1.39	63.33	79.16
Phone	0.00	2.95	4.00	38.80	55.48
Mean	0.14	5.67	1.98	48.36	57.23

(a)

	Pix2Pose Synth only(LB)	Pix2Pose+Ours	Pix2Pose+Ours+Alignment	Pix2Pose with Real Pose Labels(UB)
Ape	36.73	39.68	37.62	54.61
Bvise	59.01	72.19	78.67	90.21
Cam	27.82	56.54	65.45	72.75
Can	28.71	53.88	65.55	90.66
Cat	39.48	50.05	52.59	76.47
Drill	22.29	31.54	48.86	78.91
Duck	31.89	36.49	35.09	54.41
Eggbox	71.11	83.86	89.26	99.53
Glue	29.86	53.24	64.54	89.28
Holep	24.62	33.84	41.55	75.76
Iron	51.94	76.63	80.90	97.04
Lamp	38.73	73.35	70.74	95.59
Phone	25.73	52.13	56.47	79.56
Mean	37.53	54.88	60.56	81.14

(b)

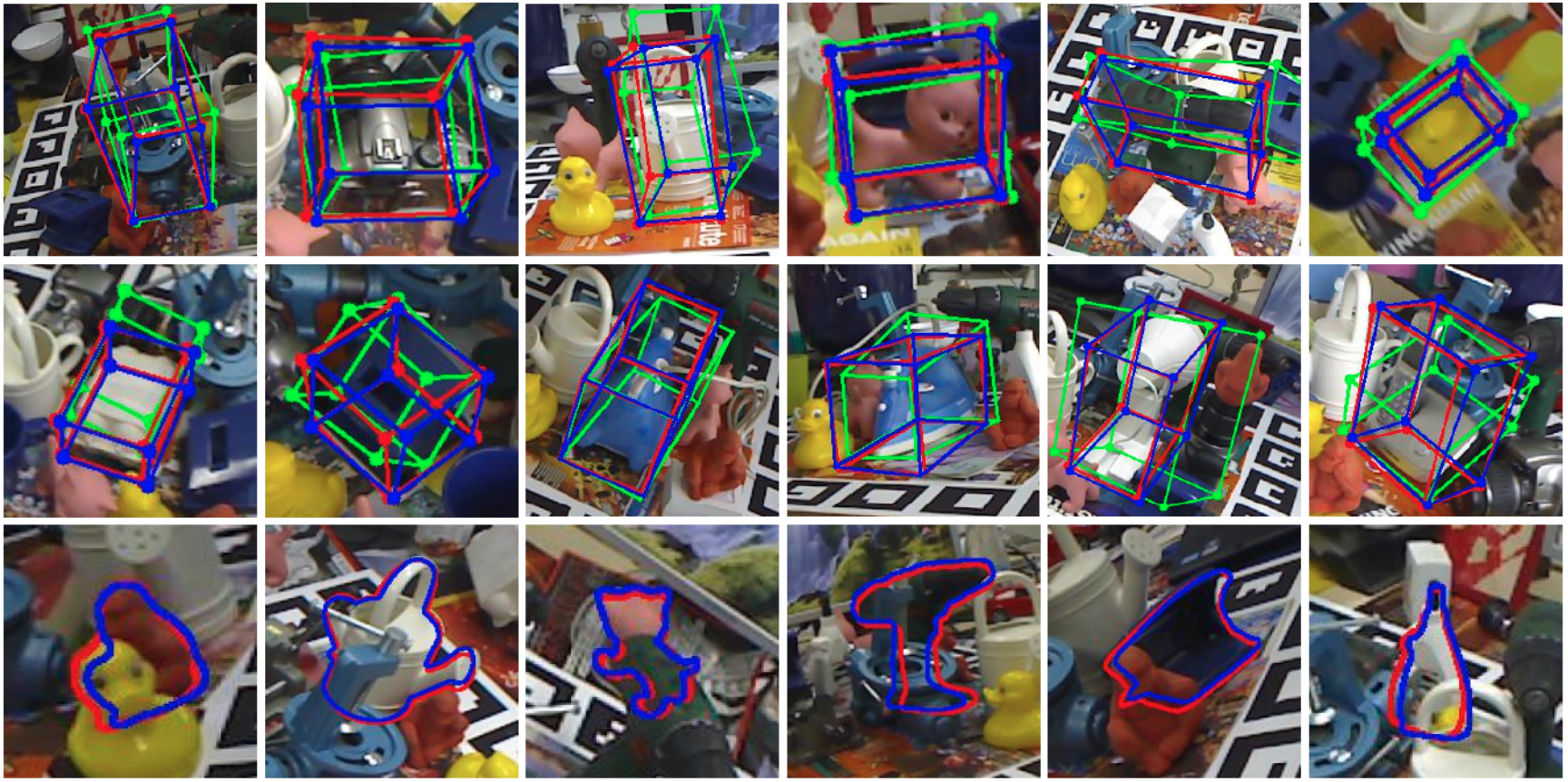


Figure 5.4: Qualitative results on LINEMOD and LINEMOD OCCLUSION. First two rows show the visualisations with projected 3D bounding boxes for LB, Ours and groundtruth on LINEMOD dataset. Red, blue and green boxes corresponds to groundtruth, self-supervised and LB respectively. The last row visualises the estimated poses for LINEMOD OCCLUSION dataset. The blue and red silhouette corresponds to the prediction and groundtruth.

Table 5.4: Comparison with state-of-the-art. Quantitative results for LINEMOD dataset by using Pix2Pose (Park et al., 2019) as the baseline pose estimator.

Train data	w/o Real Pose Labels						with Real Pose Labels		
Object	SSD	AAE	MHP	DPOD	Self6D	Ours	Tekin	DPOD	CDPN
Ape	0.0	4.2	11.9	35.1	**38.9**	37.6	21.6	53.3	**64.4**
Bvise	0.2	22.9	66.2	59.4	75.2	**78.6**	81.8	95.2	**97.8**
Cam	0.4	32.9	22.4	15.5	36.9	**65.5**	36.6	90.0	**91.7**
Can	1.4	37.0	59.8	48.8	65.6	**65.6**	68.8	94.1	**95.9**
Cat	0.5	18.7	26.9	28.1	**57.9**	52.5	41.8	60.4	**83.8**
Drill	2.6	24.8	44.6	59.3	**67.0**	48.8	63.5	**97.4**	96.2
Duck	0.0	5.9	8.3	25.6	19.6	**35.1**	27.2	66.0	**66.8**
Eggbox	8.9	81.0	55.7	51.2	**99.0**	89.2	69.6	99.6	**99.7**
Glue	0.0	46.2	54.6	34.6	**94.1**	64.5	80.0	93.8	**99.6**
HoleP	0.3	18.2	15.5	17.7	16.2	**41.5**	42.6	64.9	**85.8**
Iron	8.9	35.1	60.8	84.7	77.9	**80.9**	75.0	**99.8**	97.9
Lamp	8.2	61.2	-	45.0	68.2	**70.7**	71.1	88.1	**97.9**
Phone	0.18	36.3	34.4	20.9	58.9	**60.5**	47.7	71.4	**90.8**
Mean	2.4	32.6	38.8	40.5	59.0	**60.6**	56.0	82.6	**89.9**

5.4.2.3 *Comparison with LB and UB*

Table 5.3 (a) indicates that LB performance for the BB8 baseline is very low with ADD metric on LINEMOD. The middle rows show how unsupervised domain adaptation methods improve from the LB. Although PixelDA (Bousmalis et al., 2017) generates perceptually realistic images, it does not contribute to learn domain invariant features, as also reported in Rad et al. (2018). This can be explained by the fact that the number of real images allowed in the standard LINEMOD protocol (15%) may not be enough to learn the distribution which results in overfitting on the training data. Using GRL (Ganin et al., 2016) without real labels slightly improves the generalisability from synthetic training, as the only supervision is from the domain label, which disregards the task. On the contrary, the proposed method significantly outperforms both methods and almost reaches the performance of the UB

Table 5.5: Quantitative results on LINEMOD Occlusion. All methods in the list do not use real pose labels during training. Results except ours are from Wang et al. (2020).

Train data	RGB image only			RGB+depth
object	DPOD	CDPN	Ours	self6D
Ape	2.3	**20.0**	12.0	13.7
Can	4.0	15.1	**27.5**	43.2
Cat	1.2	**16.4**	12.0	18.7
Duck	10.5	5.0	**20.5**	32.5
Drill	7.2	22.2	**23.0**	14.4
Eggbox	4.4	**36.1**	25.1	57.8
Glue	12.9	**27.9**	27.0	54.3
HoleP	7.5	24.0	**35.0**	22.0
Mean	6.3	20.8	**22.8**	32.1

with real labels.

Table 5.3 (b) shows the performance on LINEMOD with Pix2Pose baseline. The performance with synthetic only training is much higher than BB8, and it can be explained by the fact that Pix2Pose regresses the object coordinates for every pixel and utilises RANSAC framework which makes it more robust against false estimation whereas BB8 regresses the location of eight corner points. The proposed self-supervised learning framework improves significantly from the LB, similar to the BB8 baseline experiment. The warp-align stage boosts the performance further from the self-supervised framework except for some objects such as Ape, Duck or Lamp. Alignment framework compares the texture between two images, but since those objects are almost monochromatic, alignment may not be able to provide meaningful supervision. Figure. 5.4 shows the qualitative results with LB, self-supervised and groundtruth. It shows that the alignment with groundtruth is improved after self-supervision.

5.4.2.4 *Comparison with state of the art*

In Table. 5.4, I compare the proposed work with other state-of-the-art methods. I distinguished the methods based on whether the methods are supervised with real pose labels or not. The proposed method significantly outperforms SSD (Kehl et al., 2017), AAE (Sundermeyer et al., 2018), MHP (Manhardt et al., 2019), and DPOD (Zakharov, Shugurov and Ilic, 2019) which all uses domain randomisation to learn from synthetic data. Notably, the proposed method is on par with Self6D (Wang et al., 2020), which uses both visual and geometric alignment with the utilisation of depth map during training.
Table 5.5 reports the performance on the LINEMOD OCCLUSION (Hinterstoisser et al., 2012) dataset and all methods in the table do not use real pose labels during training. The results in the table follow the BOP (Hodan et al., 2018) standard where 200 images from the original occlusion dataset are selected for the evaluation. The proposed method outperforms the DPOD (Zakharov, Shugurov and Ilic, 2019) and CDPN (Li et al., 2019) which use RGB images during training but the proposed method performs lower than Self6D (Wang et al., 2020) when tested on the images with strong occlusions since using extra modality as depth in Self6D, can help to learn more robust features in the presence of occlusions.

Table 5.6 shows the results for HomebrewedDB (Kaskman et al., 2019) dataset. HomebrewedDB is a suitable dataset to demonstrate whether the trained pose estimator is not overfitted on the data used for training and generalisable, as 3 objects from LINEMOD dataset are captured in different backgrounds and lighting conditions. The object estimator trained with 15% of unlabelled data from LINEMOD was tested on this dataset without retraining. The results show the proposed self-supervised

Table 5.6: Quantitative results on HomebrewedDB (Kaskman et al., 2019). All methods in the list do not use real pose labels during training. Results except ours are from (Wang et al., 2020).

Train data	RGB image only			RGB+depth
object	DPOD	SSD+Ref	Ours	self6D
Bvise	52.9	82.0	57.3	72.1
Drill	37.8	22.9	46.6	65.1
Phone	7.3	24.9	41.5	41.8
Mean	32.7	43.3	52.0	59.7

framework learns robust features that can generalise beyond the images in the training dataset. The proposed method outperforms DPOD (Zakharov, Shugurov and Ilic, 2019) and SSD (Kehl et al., 2017) by 59% and 20% respectively.

5.5 SUMMARY

In this chapter, I proposed a framework that consists of two main components: a self-supervised training approach and a warp-align fine-tuning stage. The experimental results show that the proposed framework can learn a robust 6D pose estimator without the use of real pose labels. Furthermore, I have also experimentally shown that the framework is flexible to use off-the-shelf object pose estimators as a baseline method by choosing two popular but distinctive approaches. The evaluations on multiple datasets show that the proposed framework largely improves from the models trained with synthetic data only, and it can generalise to the different dataset without retraining. Finally, the proposed method shows state-of-the-art performance in the absence of real data and depth information surpassing concurrent self-supervised approaches.

6

CONCLUSIONS AND FUTURE WORK

Accurate 6D object pose estimation is challenging in the presence of severe occlusion and clutters. Difficulty in capturing data and annotating pose label is another practical challenge in building robust pose estimator. This thesis proposed several algorithms each specifically tackles different challenging scenarios for 6D object pose estimation.

Chapter 1 first provided a general overview 6D object pose estimation and defined the scope of the thesis. It further introduced and categorised the challenges and motivated the contributions of this thesis.

Chapter 2 introduced three main computer vision problems that are used in this thesis to tackle the challenges. The first section provided different pipelines used and categorised the existing work in 6D object pose estimation from a single image according to how the problem is formulated. The second section introduced active vision research and reviewed existing works for different applications. The last section reviewed and discussed visual domain adaptation research for DNN.

Chapter 3 proposed a network for single-image pose estimation for the bin-picking scenario. The proposed network jointly

performed object detection, pose estimation and joint registration. A training dataset generation framework was proposed to generate data with realistic pose configurations in the bin-picking environment which enabled the proposed network to learn the relational feature. An evaluation was performed on MiSo (Multiple Instances of Single Object) setting, which means all the objects in the scene are of the same size and shape. A recent study shows that learning object pose estimation of multiple classes with a single model is difficult (Kaskman et al., 2019). An interesting future direction would be to investigate how to learn relational feature when the objects are of various size and shape.

Chapter 4 introduced an active vision system for object pose estimation for the bin-picking scenario. The problem was formulated within a reinforcement learning framework, and each component was carefully designed to extend the pose estimator beyond a single image. Although the proposed approach generated trajectory that can achieve higher pose accuracy with shorter trajectory length, the experiments were done in a synthetic environment where sensor motion planning module(as shown in Section. 2.3) is not taken into account. An interesting future direction would be to experiment with the real sensor mounted on a robotic arm. Also the episode length in the current implementation was fixed to a constant value. Potential extension of the work can include a termination action in the action space to allow early termination.

Chapter 5 proposed a novel self-supervised learning framework for 6D object pose estimation to achieve domain adaptation. The proposed approach enabled multiple off-the-shelf object pose estimator to learn only from synthetically generated training data and adapt to real data without pose labels. I compared the proposed method with several state-of-the-art baselines and

demonstrated the suitability of the approach. However, the baselines I have used in the experiments utilises separate detector, performing pose estimation on the detected cropped image of the object. For this reason, a direction for future work would be to apply unsupervised domain adaptation to a unified framework which includes both the pose estimation and detection.

6D object pose estimation has a wide range of practical application and recently it has seen increased interest from the research community. However, there are still many unsolved issues which prevents it from being used in the industry.

First challenge is the scalability issue. Instance-level object pose estimator estimate the pose of the specific object of which detailed information is known a priori, which includes texture, shape, material and size etc. Currently it is difficult to analyse how the existing models can scale to multiple instances since most of the existing methods train one neural network model per instance to maximise the performance. It is known that training multiple instances for one model severe degrade its pose estimation capability. A promising direction is the category-level pose estimation which attempts to generalise pose estimator to estimate objects in the same category.

Second challenge is the pose estimation of articulated objects. Most existing object pose estimators assumes the object is rigid which is often not realistic in the practical applications. It introduces several new challenges to the pose estimation such as higher degree of freedom(as an articulated object is composed of multiple rigid parts) and representation challenge(as various types of joints provide different physical constraints and priors for part articulations).

BIBLIOGRAPHY

Aldoma, A., Tombari, F., Di Stefano, L. and Vincze, M. (2012), A global hypotheses verification method for 3d object recognition, *in* 'European Conference on Computer Vision (ECCV)'. (pages 50, 52, 64, 65, 66, 77).

Alex, K. and Roberto, C. (2017), Geometric loss functions for camera pose regression with deep learning, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 58).

Ammirato, P., Poirson, P., Park, E., Košecká, J. and Berg, A. C. (2017), A dataset for developing and benchmarking active vision, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 73).

Arun, K. S., Huang, T. S. and Blostein, S. D. (1987), 'Least-squares fitting of two 3-d point sets', *IEEE Transactions on pattern analysis and machine intelligence* . (page 34).

Balntas, V., Doumanoglou, A., Sahin, C., Sock, J., Kouskouridas, R. and Kim, T.-K. (2017), Pose guided rgbd feature learning for 3D object pose estimation, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 33, 35, 36).

Bircher, A., Kamel, M., Alexis, K., Oleynikova, H. and Siegwart, R. (2016), Receding horizon "next-best-view" planner for 3d exploration, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (pages 40, 44).

Birdal, T. and Ilic, S. (2015), Point pair features based object detection and pose estimation revisited, *in* 'International Conference on 3D Vision (3DV)'. (pages 32, 33).

Bousmalis, K., Silberman, N., Dohan, D., Erhan, D. and Krishnan, D. (2017), Unsupervised pixel-level domain adaptation with generative adversarial networks, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 93, 96, 98, 112).

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D. and Erhan, D. (2016), Domain separation networks, *in* 'Advances in Neural Information Processing Systems (NIPS)'. (page 47).

Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J. and Rother, C. (2014), Learning 6d object pose estimation using 3D object coordinates, *in* 'European Conference on Computer Vision (ECCV)'. (pages 39, 50).

Brachmann, E., Michel, F., Krull, A., Yang, M. Y. and Gumhold, S. (2016), Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 18, 39, 107).

Brégier, R., Devernay, F., Leyrit, L. and Crowley, J. (2017), Symmetry aware evaluation of 3D object detection and pose estimation in scenes of many parts in bulk, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshop'. (pages 50, 64, 83).

Browatzki, B., Tikhanoff, V., Metta, G., Bülthoff, H. H. and Wallraven, C. (2012), Active object recognition on a humanoid robot, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 40).

Chen, B., Parra, A., Cao, J., Li, N. and Chin, T.-J. (2020), End-to-end learnable geometric vision by backpropagating pnp optimization, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 101).

Chen, T., Zhai, X., Ritter, M., Lucic, M. and Houlsby, N. (2019), Self-supervised gans via auxiliary rotation loss, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 97).

Cheng, R., Agarwal, A. and Fragkiadaki, K. (2018), 'Reinforcement learning of active vision for manipulating objects under occlusions', *Proceedings of Machine Learning Research (PMLR)* . (pages 45, 73).

Cheng, R., Wang, Z. and Fragkiadaki, K. (2018), Geometry-aware recurrent neural networks for active visual recognition, *in* 'Advances in Neural Information Processing Systems (NIPS)'. (pages 41, 44, 45).

Coates, A. and Ng, A. Y. (2010), Multi-camera object detection for robotics, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (pages 40, 73).

Collet, A. and Srinivasa, S. S. (2010), Efficient multi-view object recognition and full pose estimation, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 43).

Csurka, G. (2017), 'Domain adaptation for visual applications: A comprehensive survey', *arXiv preprint arXiv:1702.05374* . (pages 25, 45).

Cui, J., Wen, J. T. and Trinkle, J. (2019), A multi-sensor next-best-view framework for geometric model-based robotics applications, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (pages 41, 43).

D. Jayaraman and K. Grauman. (2016), Look-Ahead Before You Leap: End-to-End Active Recognition by Forecasting the Effect of Motion, *in* 'European Conference on Computer Vision (ECCV)'. (page 73).

Delmerico, J., Isler, S., Sabzevari, R. and Scaramuzza, D. (2018), 'A comparison of volumetric information gain metrics for active 3d object reconstruction', *Autonomous Robots* . (page 41).

Devin, C., Abbeel, P., Darrell, T. and Levine, S. (2018), Deep object-centric representations for generalizable robot learning, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (pages 74, 80).

Devrim Kaba, M., Gokhan Uzunbas, M. and Nam Lim, S. (2017), A reinforcement learning approach to the view planning problem, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 43).

Do, T. T., Cai, M., Pham, T. and Reid, I. D. (2018), 'Deep-6dpose: Recovering 6D object pose from a single RGB image', *arXiv preprint arXiv:1802.10367* . (pages 57, 58).

Dong, S., Xu, K., Zhou, Q., Tagliasacchi, A., Xin, S., Nießner, M. and Chen, B. (2019), 'Multi-robot collaborative dense scene reconstruction', *ACM Transactions on Graphics (TOG)* . (pages 41, 43).

Doumanoglou, A., Kouskouridas, R., Malassiotis, S. and Kim, T.-K. (2016), Recovering 6D object pose and predicting next-best-view in the crowd, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 19, 23, 32, 50, 52, 64, 71, 74, 77, 83).

Drost, B., Ulrich, M., Navab, N. and Ilic, S. (2010), Model globally, match locally: Efficient and robust 3D object recognition, *in*

'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 32, 33, 50, 64, 66, 68).

Eppner, C., Höfer, S., Jonschkowski, R., Martin, R., Sieverling, A., Wall, V. and Brock, O. (2016), Lessons from the amazon picking challenge: Four aspects of building robotic systems, *in* 'Robotics: Science and Systems (RSS)'. (pages 49, 71).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M. and Lempitsky, V. (2016), 'Domain-adversarial training of neural networks', *Journal of Machine Learning Research (JMLR)* . (pages 47, 93, 96, 97, 112).

Georgakis, G., Karanam, S., Wu, Z. and Kosecka, J. (2018), 'Matching rgb images to cad models for object pose estimation', *arXiv preprint arXiv:1811.07249* . (page 96).

Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D. and Li, W. (2016), Deep reconstruction-classification networks for unsupervised domain adaptation, *in* 'European Conference on Computer Vision (ECCV)'. (page 47).

Godard, C., Mac Aodha, O., Firman, M. and Brostow, G. J. (2019), Digging into self-supervised monocular depth estimation, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 95, 98).

Grtner, E., Pirinen, A. and Sminchisescu, C. (2020), Deep reinforcement learning for active human pose estimation., *in* 'Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)'. (page 40).

Gupta, S., Arbelaez, P., Girshick, R. and Malik, J. (2015), Aligning 3d models to rgb-d images of cluttered scenes, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 36, 37).

Han, X., Zhang, Z., Du, D., Yang, M., Yu, J., Pan, P., Yang, X., Liu, L., Xiong, Z. and Cui, S. (2019), 'Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image'. (page 41).

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T.-Y. and Ma, W.-Y. (2016), Dual learning for machine translation, *in* 'Advances in Neural Information Processing Systems (NIPS)'. (page 48).

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017), Mask r-cnn, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 34).

Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N. and Lepetit, V. (2011), Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 18, 19, 35, 50, 64, 66, 68).

Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K. and Navab, N. (2012), Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes, *in* 'Asian Conference on Computer Vision (ACCV)'. (pages 30, 65, 81, 85, 106, 114).

Hochreiter, S. and Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* . (page 52).

Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X. and others (2018), Bop: benchmark for 6d object pose estimation, *in* 'European Conference on Computer Vision (ECCV)'. (page 114).

Hodaň, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M. and Zabulis, X. (2017), 'T-less: An rgb-d dataset for 6d pose estimation of texture-less objects', *Proceedings of the IEEE Winter*

Conference on Applications of Computer Vision (WACV) . (pages 19, 22).

Hodaň, T., Zabulis, X., Lourakis, M., Obdržálek, Š. and Matas, J. (2015), 'Detection and fine 3d pose estimation of texture-less objects in RGB-D images', *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* . (pages 22, 35).

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A. and Darrell, T. (2018), 'Cycada: Cycle-consistent adversarial domain adaptation'. (page 97).

Holz, D., Nieuwenhuisen, M., Droeschel, D., Stückler, J., Berner, A., Li, J., Klein, R. and Behnke, S. (2014), Active recognition and manipulation for mobile robot bin picking, *in* 'Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe:', Springer, pp. 133–153. (page 19).

Hosang, J., Benenson, R. and Schiele, B. (2017), Learning non-maximum suppression, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 50, 51, 52, 54, 60).

Hu, H., Gu, J., Zhang, Z., Dai, J. and Wei, Y. (2018), Relation networks for object detection, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 50, 52, 54).

Hu, J., Lu, J. and Tan, Y.-P. (2015), Deep transfer metric learning, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 46).

Ionescu, C., Papava, D., Olaru, V. and Sminchisescu, C. (2014), 'Human3.6m: Large scale datasets and predictive methods for

3d human sensing in natural environments', *IEEE Transactions on Pattern Analysis and Machine Intelligence* . (page 95).

Isler, S., Sabzevari, R., Delmerico, J. and Scaramuzza, D. (2016), An information gain formulation for active volumetric 3d reconstruction, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 42).

Jayaraman, D. and Grauman, K. (2016), Look-ahead before you leap: End-to-end active recognition by forecasting the effect of motion, *in* 'European Conference on Computer Vision (ECCV)'. (page 40).

Johns, E., Leutenegger, S. and Davison, A. J. (2016), Pairwise decomposition of image sequences for active multi-view recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 40).

Kaskman, R., Zakharov, S., Shugurov, I. and Ilic, S. (2019), Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshop'. (pages 23, 30, 114, 115, 118).

Kato, H., Ushiku, Y. and Harada, T. (2018), Neural 3d mesh renderer, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 101, 107).

Kehl, W., Manhardt, F., Tombari, F., Ilic, S. and Navab, N. (2017), SSD-6D: Making rgb-based 3D detection and 6D pose estimation great again, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 22, 33, 34, 36, 50, 57, 59, 66, 67, 68, 96, 114, 115).

Kendall, A., Grimes, M. and Cipolla, R. (2015), Posenet: A convolutional network for real-time 6-dof camera relocalization, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 37).

Kiciroglu, S., Rhodin, H., Sinha, S., Salzmann, M. and Fua, P. (2020), Activemocap: Optimized viewpoint selection for active human motion capture, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 40, 44).

Kim, J. and Canny, J. (2017), Interpretable learning for self-driving cars by visualizing causal attention, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 74).

Kim, T., Cha, M., Kim, H., Lee, J. and Kim, J. (2017), 'Learning to discover cross-domain relations with generative adversarial networks', *International Conference on Machine Learning (ICML)* . (page 47).

Kingma, D. P. and Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* . (page 107).

Konishi, Y., Hanzawa, Y., Kawade, M. and Hashimoto, M. (2016), Fast 6d pose estimation from a monocular image using hierarchical pose trees, *in* 'European Conference on Computer Vision (ECCV)'. (page 35).

Kostrikov, I. (2018), 'Pytorch implementations of reinforcement learning algorithms', `https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail`. (page 84).

Krainin, M., Curless, B. and Fox, D. (2011), Autonomous generation of complete 3d object models using next best view manipulation planning, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 40).

Kriegel, S., Brucker, M., Marton, Z., Bodenmüller, T. and Suppa, M. (2013), Combining object modeling and recognition for

active scene exploration, *in* 'Proceedings of theIEEE/RSJ International Conference on Intelligent Robots andSystems (IROS)'. (page 41).

Kriegel, S., Rink, C., Bodenmüller, T., Narr, A., Suppa, M. and Hirzinger, G. (2012), Next-best-scan planning for autonomous 3d modeling, *in* '2012 IEEE/RSJ International Conference on Intelligent Robots and Systems'. (page 40).

Krull, A., Brachmann, E., Michel, F., Yang, M. Y., Gumhold, S. and Rother, C. (2015), Learning analysis-by-synthesis for 6d pose estimation in rgb-d images, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 50).

Kulkarni, N., Gupta, A. and Tulsiani, S. (2019), Canonical surface mapping via geometric cycle consistency, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 98).

Kumra, S. and Kanan, C. (2017), Robotic grasp detection using deep convolutional neural networks, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)'. (page 19).

Lee, D., Cha, G., Yang, M.-H. and Oh, S. (2016), Individualness and determinantal point processes for pedestrian detection, *in* 'European Conference on Computer Vision (ECCV)'. (page 52).

Lee, S., Kim, D., Kim, N. and Jeong, S.-G. (2019), Drop to adapt: Learning discriminative features for unsupervised domain adaptation, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 97).

Li, B. (2017), 3d fully convolutional network for vehicle detection in point cloud, *in* 'Proceedings of the IEEE/RSJ International

Conference on Intelligent Robots and Systems (IROS)'. (page 38).

Li, Y., Wang, G., Ji, X., Xiang, Y. and Fox, D. (2018), Deepim: Deep iterative matching for 6d pose estimation, *in* 'European Conference on Computer Vision (ECCV)'. (page 34).

Li, Z., Wang, G. and Ji, X. (2019), Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 33, 39, 93, 114).

Lindlbauer, D. and Wilson, A. (2018), Remixed reality: Manipulating space and time in augmented reality, *in* 'ACM CHI Conference on Human Factors in Computing Systems'. (page 19).

Liu, L., Lu, J., Xu, C., Tian, Q. and Zhou, J. (2019), Deep fitting degree scoring network for monocular 3d object detection, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 19).

Liu, L., Xia, X., Sun, H., Shen, Q., Xu, J., Chen, B., Huang, H. and Xu, K. (2018), 'Object-aware guidance for autonomous scene reconstruction', *ACM Transactions on Graphics (Proc. SIGGRAPH)* . (page 40).

Liu, M.-Y. and Tuzel, O. (2016), Coupled generative adversarial networks, *in* 'Advances in Neural Information Processing Systems (NIPS)'. (page 46).

Liu, M.-Y., Tuzel, O., Veeraraghavan, A., Taguchi, Y., Marks, T. and Chellappa, R. (2012), 'Fast object localization and pose estimation in heavy clutter for robotic bin picking', *The International Journal of Robotics Research (IJRR)* . (page 52).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. (2016), Ssd: Single shot multibox detector, *in* 'European Conference on Computer Vision (ECCV)'. (pages 33, 36).

Long, M., Zhu, H., Wang, J. and Jordan, M. I. (2016), Unsupervised domain adaptation with residual transfer networks, *in* 'Advances in Neural Information Processing Systems (NIPS)'. (page 46).

Mahendran, S., Ali, H. and Vidal, R. (2017), 3d pose regression using convolutional neural networks, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshop'. (page 37).

Manhardt, F., Arroyo, D. M., Rupprecht, C., Busam, B., Birdal, T., Navab, N. and Tombari, F. (2019), Explaining the ambiguity of object detection and 6d pose from visual data, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 114).

Manhardt, F., Kehl, W., Navab, N. and Tombari, F. (2018), Deep model-based 6d pose refinement in rgb, *in* 'European Conference on Computer Vision (ECCV)'. (page 34).

Marion, P., Florence, P. R., Manuelli, L. and Tedrake, R. (2018), Label fusion: A pipeline for generating ground truth labels for real rgbd data of cluttered scenes, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 21).

Maver, J., Leonardis, A. and Solina, F. (1993), Planning the next view using the max-min principle, *in* 'International Conference on Computer Analysis of Images and Patterns'. (page 42).

Mees, O., Tatarchenko, M., Brox, T. and Burgard, W. (2019), Self-supervised 3d shape and viewpoint estimation from single

images for robotics, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)'. (page 97).

Mendoza, M., Vasquez-Gomez, J., Taud, H., Sucar, L. and Reta, C. (2019), 'Supervised learning of the next-best-view for 3d object reconstruction', *Pattern Recognition Letters* . (page 41).

Motiian, S., Piccirilli, M., Adjeroh, D. A. and Doretto, G. (2017), Unified deep supervised domain adaptation and generalization, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 46).

Munoz, E., Konishi, Y., Murino, V. and Del Bue, A. (2016), Fast 6d pose estimation for texture-less objects from a single rgb image, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 35).

Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R. and Kim, K. (2018), Image to image translation for domain adaptation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 97).

Mustafa, W., Pugeault, N. and Kruger, N. (2013), Multi-view object recognition using view-point invariant shape relations and appearance information, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 73).

Oberweger, M., Rad, M. and Lepetit, V. (2018), Making deep heatmaps robust to partial occlusions for 3d object pose estimation, *in* 'European Conference on Computer Vision (ECCV)'. (page 38).

Palazzi, A., Bergamini, L., Calderara, S. and Cucchiara, R. (2018), End-to-end 6-dof object pose estimation through differentiable

rasterization, *in* 'European Conference on Computer Vision (ECCV)'. (page 97).

Papon, J. and Schoeler, M. (2015), Semantic pose using deep networks trained on synthetic rgb-d, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 36, 37).

Park, K., Patten, T. and Vincze, M. (2019), Pix2pose: Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 18, 22, 23, 93, 95, 99, 106, 112).

Park, K., Prankl, J. and Vincze, M. (2017), Mutual hypothesis verification for 6D pose estimation of natural objects, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops on Recovering 6D Object Pose'. (pages 50, 52).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. (2017), 'Automatic differentiation in pytorch', *Advances in Neural Information Processing Systems Workshop* . (page 107).

Pavlakos, G., Zhou, X., Chan, A., Derpanis, K. G. and Daniilidis, K. (2017), 6-dof object pose from semantic keypoints, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 38).

Payet, N. and Todorovic, S. (2011), From contours to 3d object detection and pose estimation, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 35).

Peng, S., Liu, Y., Huang, Q., Bao, H. and Zhou, X. (2019), Pvnet: Pixel-wise voting network for 6dof pose estimation, *in* 'Proceed-

ings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 23, 33, 38, 107).

Poirson, P., Ammirato, P., Fu, C.-Y., Liu, W., Kosecka, J. and Berg, A. C. (2016), Fast single shot detection and pose estimation, *in* 'International Conference on 3D Vision (3DV)'. (page 36).

Rad, M. and Lepetit, V. (2017), Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 18, 33, 38, 50, 93, 95, 99, 106, 107).

Rad, M., Oberweger, M. and Lepetit, V. (2018), Feature mapping for learning fast and accurate 3d pose inference from synthetic images, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 93, 96, 112).

Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J. and Black, M. J. (2019), Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 95, 98).

Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J. and Gkioxari, G. (2020), 'Accelerating 3d deep learning with pytorch3d', *arXiv preprint arXiv:2007.08501* . (pages 101, 107).

Redmon, J. and Farhadi, A. (2017), Yolo9000: Better, faster, stronger, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 33).

Ren, S., He, K., Girshick, R. and Sun, J. (2015), Faster r-cnn: Towards real-time object detection with region proposal net-

works, *in* 'Advances in Neural Information Processing Systems (NIPS)'. (pages 33, 57, 107).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A. and Bernstein, M. (2015), 'Imagenet large scale visual recognition challenge', *International Journal of Computer Vision (IJCV)* . (pages 17, 43, 68).

Russo, P., Carlucci, F. M., Tommasi, T. and Caputo, B. (2018), From source to target and back: symmetric bi-directional adaptive gan, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 46).

Sahin, C., Garcia-Hernando, G., Sock, J. and Kim, T.-K. (2020), 'A review on object pose recovery: from 3d bounding box detectors to full 6d pose estimators', *Image and Vision Computing* . (page 31).

Sahin, C. and Kim, T.-K. (2018), Category-level 6d object pose recovery in depth images, *in* 'European Conference on Computer Vision Workshop', pp. 0–0. (page 20).

Sahin, C., Kouskouridas, R. and Kim, T.-K. (2017), A learning-based variable size part extraction architecture for 6D object pose recovery in depth, *in* 'Journal of Image and Vision Computing (IVC)'. (pages 66, 68).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017), 'Proximal policy optimization algorithms', *arXiv preprint arXiv:1707.06347* . (pages 82, 84).

Simonyan, K. and Zisserman, A. (2015), Very deep convolutional networks for large-scale image recognition, *in* 'International Conference on Learning Representations (ICLR)'. (page 104).

Stewart, R., Andriluka, M. and Ng, A. (2016), End-to-end people detection in crowded scenes, *in* 'Proceedings of the IEEE Con-

ference on Computer Vision and Pattern Recognition (CVPR)'. (page 52).

Su, H., Qi, C. R., Li, Y. and Guibas, L. J. (2015), Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 37).

Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B. and Freeman, W. T. (2018), Pix3d: Dataset and methods for single-image 3d shape modeling, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 95).

Sundermeyer, M., Marton, Z.-C., Durner, M., Brucker, M. and Triebel, R. (2018), Implicit 3d orientation learning for 6d object detection from rgb images, *in* 'European Conference on Computer Vision (ECCV)'. (pages 22, 33, 35, 36, 93, 96, 97, 114).

Susanto, W., Rohrbach, M. and Schiele, B. (2012), '3D Object Detection With Multiple Kinects', *European Conference on Computer Vision (ECCV)* . (page 73).

Sutton, R. S., McAllester, D. A., Singh, S. P. and Mansour, Y. (2000), Policy gradient methods for reinforcement learning with function approximation, *in* 'Advances in Neural Information Processing Systems (NIPS)'. (page 44).

Tejani, A., Kouskouridas, R., Doumanoglou, A., Tang, D. and Kim, T. K. (2018), 'Latent-class hough forests for 6DoF object pose estimation', *IEEE Transactions on Pattern Analysis and Machine Intelligence* . (page 68).

Tejani, A., Tang, D., Kouskouridas, R. and Kim, T.-K. (2014), Latent-class hough forests for 3D object detection and pose estimation, *in* 'European Conference on Computer Vision (ECCV)'. (pages 19, 23, 32, 33, 50, 64, 66).

Tekin, B., Sinha, S. N. and Fua, P. (2018), Real-time seamless single shot 6d object pose prediction, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 33, 38, 50, 93).

Tjaden, H., Schwanecke, U. and Schomer, E. (2017), Real-time monocular pose estimation of 3d objects using temporally consistent local color histograms, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)', pp. 124–132. (page 35).

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W. and Abbeel, P. (2017), Domain randomization for transferring deep neural networks from simulation to the real world, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)'. (page 96).

Tombari, F., Franchi, A. and Di Stefano, L. (2013), Bold features to detect texture-less objects, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (page 22).

Tran, N.-T., Tran, V.-H., Nguyen, N.-B. and Cheung, N.-M. (2019), 'An improved self-supervised gan via adversarial training', *arXiv preprint arXiv:1905.05469* . (page 97).

Tulsiani, S. and Malik, J. (2015), Viewpoints and keypoints, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 38).

Tzeng, E., Hoffman, J., Saenko, K. and Darrell, T. (2017), Adversarial discriminative domain adaptation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 47).

Ulrich, M., Wiedemann, C. and Steger, C. (2011), 'Combining scale-space and similarity-based aspect graphs for fast 3d object

recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence* . (page 35).

Walas, K., Hodan, T., Kim, T., Matas, J., Rother, C., Michel, F., Lepetit, V., Leonardis, A., Stegger, C., Sahin, C. and Kouskouridas, R. (2017), '3rd international workshop on recovering 6D object pose'. http://cmp.felk.cvut.cz/sixd/workshop_2017/ (accessed on: 2018-05-05) (page 49).

Wan, C., Probst, T., Gool, L. V. and Yao, A. (2019), Self-supervised 3d hand pose estimation through training by fitting, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 94, 97, 99).

Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L. and Savarese, S. (2019), Densefusion: 6d object pose estimation by iterative dense fusion, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 34).

Wang, D., Devin, C., Cai, Q.-Z., Yu, F. and Darrell, T. (2019), 'Deep object centric policies for autonomous driving'. (pages 31, 80).

Wang, G., Manhardt, F., Shao, J., Ji, X., Navab, N. and Tombari, F. (2020), Self6d: Self-supervised monocular 6d object pose estimation, *in* 'European Conference on Computer Vision (ECCV)'. (pages 94, 97, 99, 106, 113, 114, 115).

Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S. and Guibas, L. J. (2019), 'Normalized object coordinate space for category-level 6d object pose and size estimation', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* . (pages 33, 39).

Wohlhart, P. and Lepetit, V. (2015), Learning descriptors for object recognition and 3D pose estimation, *in* 'Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 35, 36).

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J. (2015), 3d shapenets: A deep representation for volumetric shapes, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 41, 44, 45).

Xu, K., Shi, Y., Zheng, L., Zhang, J., Liu, M., Huang, H., Su, H., Cohen-Or, D. and Chen, B. (2016), '3d attention-driven depth acquisition for object identification', *ACM Transactions on Graphics (TOG)* . (page 74).

Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y. and Zuo, W. (2017), Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 46).

Yuan, S., Ye, Q., Stenger, B., Jain, S. and Kim, T.-K. (2017), Big-hand2. 2m benchmark: Hand pose dataset and state of the art analysis, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 95).

Zakharov, S., Kehl, W. and Ilic, S. (2019), Deceptionnet: Network-driven domain randomization, *in* 'Proceedings of the IEEE International Conference on Computer Vision (ICCV)'. (pages 96, 97).

Zakharov, S., Kehl, W., Planche, B., Hutter, A. and Ilic, S. (2017), 3d object instance recognition and pose estimation using triplet loss with dynamic margin, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)'. (page 35).

Zakharov, S., Shugurov, I. and Ilic, S. (2019), Dpod: 6d pose object detector and refiner, *in* 'Proceedings of the IEEE International

Conference on Computer Vision (ICCV)'. (pages 33, 34, 39, 93, 96, 114, 115).

Zeng, A., Yu, K.-T., Song, S., Suo, D., Walker, E., Rodriguez, A. and Xiao, J. (2017), Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (pages 25, 52, 71).

Zeng, R., Wen, Y., Zhao, W. and Liu, Y.-J. (2020), 'View planning in robot active vision: A survey of systems, algorithms, and applications', *Computational Visual Media* . (page 31).

Zhang, H. and Cao, Q. (2017), 'Texture-less object detection and 6d pose estimation in rgb-d images', *Robotics and Autonomous Systems (RAS)* . (page 22).

Zhang, R., Isola, P., Efros, A. A., Shechtman, E. and Wang, O. (2018), The unreasonable effectiveness of deep features as a perceptual metric, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 104).

Zhao, H., Gallo, O., Frosio, I. and Kautz, J. (2017), 'Loss functions for image restoration with neural networks', *IEEE Transactions on Computational Imaging* . (page 105).

Zhou, X., Karpur, A., Luo, L. and Huang, Q. (2018), Starmap for category-agnostic keypoint and viewpoint estimation, *in* 'Proceedings of the European Conference on Computer Vision (ECCV)', pp. 318–334. (page 38).

Zhou, Y., Barnes, C., Jingwan, L., Jimei, Y. and Hao, L. (2019), On the continuity of rotation representations in neural networks, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (page 18).

Zhu, J.-Y., Park, T., Isola, P. and Efros, A. A. (2017), Unpaired image-to-image translation using cycle-consistent adversarial networks, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'. (pages 47, 94).

Zhu, M., Derpanis, K. G., Yang, Y., Brahmbhatt, S., Zhang, M., Phillips, C., Lecce, M. and Daniilidis, K. (2014), Single image 3d object detection and pose estimation for grasping, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)'. (page 19).

Zou, Y., Luo, Z. and Huang, J.-B. (2018), Df-net: Unsupervised joint learning of depth and flow using cross-task consistency, *in* 'European Conference on Computer Vision (ECCV)'. (page 98).