How Geometry Meets Learning in Pose Estimation

THE UNIVERSITY of ADELAIDE

Ming Cai

School of Computer Science

University of Adelaide

A thesis submitted for the degree of

Doctor of Philosophy

Supervised by:

Prof. Ian D. Reid

Prof. Chunhua Shen

July 2020

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signature:

Date: 8/7/2020

Acknowledgements

Personal

The year spent with this thesis have led to my incurring many debts of gratitude, and it is my pleasure to acknowledge these now.

My thesis has gained much from the supervision of Professor Ian D. Reid, who have been encouraging and inspiring during my whole Ph.D. His creative idea opened doors for my study and research, and his patience and trust kept me confident during those down times. What I have learnt from him is not only the answers and solutions to research questions, but also the way of conducting research and collaborating with others, and most importantly, how to deal with difficulties. The experience working with him is a treasure to me, which will keep guiding the way of research in the rest of my career.

The advice of my co-supervisor, Professor Chunhua Shen were also inspiring. The discussions with him, which led to the ideas in this thesis, are very much appreciated.

Without the care of my lab mates and postdocs I could not have completed this thesis. My friends, Huangying Zhan, Kejie Li, Chamara Saroj Weerasekera, Vladimir Nekrasov, Rafael Felix, Mehdi Hosseinzadeh, Tong Shen have taken good care of me both in study and daily life during these years. The time spent with Shin Fang Ch'ng, Chee Kheng Chng, Violetta Shevchenko, Zhipeng Cai, Gabriel Maicas, Samya Bagchi, Ergnoor Shehu, Jiawang Bian, Zhi Tian, Tong He and Hao Chen will alway be a part of my memory.

Last but not least, I would like to thank my parents who have been very supportive throughout my Ph.D career.

Institutional

Much of that time has been spent in the School of Computer Science of University of Adelaide and in using their varied resources. Australian Centre for Robotic Vision provided huge support for me to complete my Ph.D study. I am very grateful to those opportunity working with people from other nodes. I would like also thank the Australian Institute for Machine Learning for providing new working space and facilities at the end of my Ph.D.

Abstract

This thesis focuses on one of the fundamental problems in computer vision, six-degree-of-freedom (6dof) pose estimation, whose task is to predict the geometric transformation from the camera to a target of interest, from only RGB inputs. Solutions to this problem have been proposed using the technique of image retrieval or sparse 2D-3D correspondence matching with geometric verification. Thanks to the development of deep learning, the direct regression-based (compute pose directly from image-to-pose regression) and indirect reconstruction-based (solve pose via dense matching between image and 3D reconstruction) approaches using neural network recently draw growing attention in community. Although models have been proposed for both camera relocalisation and object pose estimation using a deep network base, there are still open questions. In this thesis, we investigate several problems in pose estimation regarding end-to-end object pose inference, uncertainty of pose estimation in regression-based method and self-supervision for reconstruction-based learning both for scenes and objects.

We focus on the end-to-end 6dof pose regression for objects in the first part of this thesis. Traditional methods that predict the 6dof pose for objects usually rely on the 3D CAD model and require a multi-step scheme to compute the pose. We alternatively use the idea of direct pose regression for objects based on a region proposed network Mask R-CNN, which is well-known for object detection and instance segmentation. Our newly proposed network head regresses a 4D vector from the RoI feature map of each object. A 3D vector from Lie algebra is used as the representation for rotation. Another one scalar for the z-axis of translation is predicted to recover the full 3D translation along with the position of bounding boxes. This simplification avoids the spatial ambiguity for object in the scope of 2D image caused by RoIPooling. Our method performs accurately at inference time, and faster than methods that require 3D models and refinement in their pipeline.

We estimate the uncertainty for the pose regressed by a deep model in the second part. A CNN is combined with Gaussian Process Regression (GPR) to build a framework that directly obtains a predictive distribution over camera pose. The combination is achieved by exploiting the CNN to extract discriminative features and using the GPR to perform probabilistic inference. In order to prevent the complexity of uncertainty estimation from growing with the number of training

images in the datasets, we use pseudo inducing CNN feature points to represent the whole dataset and learn their representations using Stochastic Variational Inference (SVI). This makes GPR a parametric model, which can be learnt together with the CNN backbone at the same time. We test the proposed hybrid framework on the problem of camera relocalisation.

The third and fourth parts of our thesis have similar objectives: seeking self-supervision for the learning of dense reconstruction for pose estimation from images without using the ground truth 3D model of scenes (in part 3) and objects (in part 4). We explore an alternative supervisory signal from multi-view geometry. Photometric and/or featuremetric consistency in image pairs from different viewpoints is proposed to constrain the learning of the world-centric coordinates (part 3) and object-centric coordinates (part 4). The dense reconstruction model is subsequently used as 2D-3D correspondences establisher at inference time to compute the 6dof pose using PnP plus RANSAC. Our 3D model free methods for pose estimation eliminate the dependency on 3D models used in state-of-the-art approaches.

Publications

This thesis contains the following work that has been published or prepared for publication:

- Thanh-Toan Do, Trung Pham, **Ming Cai**, and Ian Reid (2018). "Real-time monocular object instance 6d pose estimation". In: British Machine Vision Conference;
- **Ming Cai**, Chunhua Shen, and Ian D Reid (2018). "A Hybrid Probabilistic Model for Camera Relocalization." In: British Machine Vision Conference;
- **Ming Cai**, Huangying Zhan, Chamara Saroj Weerasekera, Kejie Li, and Ian Reid (2019). "Camera Relocalization by Exploiting Multi-View Constraints for Scene Coordinates Regression". In: Proceedings of the IEEE International Conference on Computer Vision Workshops;
- **Ming Cai** and Ian Reid (2020). "Reconstruct Locally, Localize Globally: A Model Free Method for Object Pose Estimation". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Contents

List of Figures

List of Tables

1 Introduction

Contents

This chapter addresses the focused tasks of thesis and the motivation behind it. We also detail the objectives, methods and the main contributions of our work.

1.1 Introduction

The perception of a physical world is a time-honored topic in robotics. As part of the autonomous system, it purveys important information for the subsequent control and/or navigation modules, making smart agents – such as robots and unmanned vehicles (we use robots as the representative for the rest of this thesis) – operate freely in the surrounding environment.

The system for spatial awareness is expected to deliver two major outcomes: interpretation of *what* things are in the environment and robot's *relationship* to them.

Thanks to the growing interest in camera sensors and deep learning, computer vision community has recently gained massive understanding of the composition of the environment from images by the means of classification (Deng et al. 2009; Krizhevsky et al. 2012; He et al. 2015; He et al. 2016), detection (Girshick 2015; Ren et al. 2015; Redmon et al. 2016; Liu et al. 2016; He et al. 2017) and semantic or instance segmentation (Long et al. 2015; Ronneberger et al. 2015; He et al. 2017). What is however interesting is that in the early years of computer vision, researchers actually paid more attention to geometric perception, *e.g.* 3D reconstruction (Van Heel 1987; Whitaker 1998; Cline et al. 1987), shape registration (Besl et al. 1992; Bajura et al. 1995; Blais et al. 1995), motion tracking (Faugeras et al. 1988; Dorfmüller 1999) and pose estimation (Haralick et al. 1989; Lu et al. 1997; Dementhon et al. 1995), with applications in Virtual Reality (VR), Augment Reality (AR) and autonomous driving.

One of the goals of geometric perception is to build a relationship between a robot and the world. Such relationship is commonly represented by a six-degree-of-freedom (6dof) pose, *i.e.*, position and orientation. It quantitatively provides spatial information for robots to physically interact with the world. For instance, a robot should know its position with respect to a map in order to plan its next move, and be aware of the orientations of nearby objects for manipulation.

Having 3D sensors such as a depth camera (Covell et al. 2006; Ye et al. 2011) or LIDAR (Hess et al. 2016; Wolcott et al. 2014) makes it easy to infer geometry, but the high cost and relatively poor large scale measurement prevent them from being universal solutions. Thanks to the compactness, low cost and the nature of passive sensing, pose estimation from optical camera has always been an active topic in computer vision.

Recovering the pose from scratch by using RGB images only is, however, challenging. The 6dof pose is strictly defined as the transformation between two coordinate frames, but an image is not even close to an explicit and direct

representation of such transformation. Therefore, this problem is commonly formulated following the concept of retrieval[1]. One version of such retrieval is dataset-based, in which a database – serves as the prior of the interested scene/target – that contains images and their well-calibrated poses is created before pose inference takes place. The goal of pose estimation is to compute a novel pose for a previously unseen image, exploiting information in the database. Another version is model-based, which assumes the existence of a 3D model (*e.g.* CAD model) and the task then is to match features to retrieve for pose.

From a geometric perspective, a pose mathematically describes the 6dof transformation of the capturing camera with respect to a reference system. The definition of pose estimation varies depending on the characteristics of reference. We consider two such references in this thesis: a static world and certain interest objects. More specifically, we perform two tasks in our thesis: i) relocalising a moving camera in a previously-known scene, and ii) estimating the poses of known objects that move (relatively) with respect to the camera.

1.2 Background

The classical solutions adopted by researchers to geometry estimation can be commonly categorized into two groups: retrieval-based and correspondences-based. Namely, retrieval-based methods find the best matched image (or images) in the training set against the query image, and determine the new pose based on that of the matching (or matchings). In correspondence-based methods, features such as points or lines are matched between image and the object or the scene to solve the pose using a subsequent geometric verification. These methods provide a gold

[1] There is another line of pose estimation that incrementally computes the pose of camera (or target) from sequential images by frame-to-frame tracking, *e.g.* a visual odometer. Although with similar objective, the method of pose tracker differs from the interests of this thesis, therefore we choose not to mention them extensively.

standard for pose estimation and their effectiveness has been demonstrated both in indoor and outdoor environments.

Nonetheless, methods in both families usually require to detect sparse image features to conduct further pose determination. In retrieval-based methods, they are combined altogether to characterize the image globally, whereas in correspondence-based methods, the establishment of correspondences is commonly formulated as a descriptor matching problem that is solved using nearest neighbor search between features in 2D and 3D (Sattler 2013). Therefore, the community has particularly investigated the representation of the sparse feature. Descriptors such as SIFT (Lowe 1999), SURF (Bay et al. 2006) and ORB (Rublee et al. 2011) summarize the surrounding appearance of certain key-points from a corner detector based on the information on the RGB images. They perform very well in terms of points matching and are used extensively in almost every geometric computer vision task.

However, due to the limitations of the sparse feature, such as sparsity, lack of global information and sensitiveness to dynamics, they can be deployed only in some constrained environments. The generalization and robustness of these methods to real world applications – where the images suffer from motion blur, regions without textures and change of light and/or weather condition – has always been under the eyes of researchers.

What has been achieved in those content recognition tasks suggests that, using Convolutional Neural Networks (CNNs) (Krizhevsky et al. 2012) overcomes the short-comings of the previous solutions for recognition that rely on local and discriminative image information. These nested convolutional layers extract dense feature maps in different scales for per-pixel tasks such as segmentation. The feature map also can be flattened to a feature vector to describe the high-level, global and task-driven knowledge of the image for tasks like classification and place recognition. Due to the powerful modeling ability from large amounts of data, the advantages of

CNNs include but are not limited to: multi-scale feature fusion, generalization to category, end-to-end inference, per-pixel dense prediction (if needed) *etc*.

As a matter of fact, tasks that benefit materially from CNNs commonly focus on finding out *what* are in the image, despite the individual peculiarity such as different instances, arrangements and viewpoints. In other words, one property the CNNs can offer is the generalization. Consequently, most of the models are therefore designed and trained to be invariant to changes in *relationship*. For example, the class of an object should be consistent no matter from which viewpoint the object is captured.

The behavior of CNNs in classification-like tasks may advise that they are prone to generalizing by ignoring relationship. They potentially sacrifice the ability to recover pose specific information while achieving viewpoint invariance. Therefore two significant question marks have not been cleared yet that it is not clear whether are CNNs good at modeling relationship or not, and if yes, it is not well understood how to model this relative information from CNNs. With these two questions remain open, several attempts have been made to use the structure of CNNs for the problem of 6dof pose estimation.

Based on the type of output from CNNs, we group the existing learning-based methods into two categories: regression-based method and reconstruction-based method. Regression-based methods use neural networks to perform direct pose regression from an input image, where the model is learned by mapping the training images to their ground truth 6dof poses. Representatives of this family are PoseNet (Kendall et al. 2015) and its variants (Kendall et al. 2016; Kendall et al. 2017). As for reconstruction-based methods(Shotton et al. 2013; Brachmann et al. 2014; Valentin et al. 2015; Brachmann et al. 2017; Brachmann et al. 2018), the key idea is to learn a mapping from image pixels to their 3D coordinates in a reference space (which is world-centric coordinate space for the task of camera relocalisation and object-centric coordinate space for object pose estimation). The 6dof pose is then solved based on the dense 2D-3D correspondence provided by the networks.

Interestingly, although using different tools to formulate the problem, the deep learning based modern methods behave conceptually alike with the classical methods. The reason regression-based methods are believed to be similar with the retrieval-based method is that the pose regressors essentially determine a test pose by interpolating from training poses. Meanwhile, at inference time of reconstruction-based methods, the model is applied to a test image to conduct 2D-to-3D regression densely and use these correspondences to compute the pose, acting just like the correspondences-based methods. However, instead of building an explicit 3D map to allow descriptor matching, the process of finding correspondences between points and their 3D coordinates is carried by training CNNs to perform dense 3D reconstruction for each image.

1.3 Motivation and Objectives

Stemming from the success achieved by aforementioned modern methods, the overall objective of our thesis is also to develop learning-based methods to solve the traditional 6dof pose estimation tasks. Though this aim in itself is not novel, we address several shortcomings of previous work in this thesis, which we believe are the factors that can be solved to improve the completeness of the learning-based method for pose estimation, and provide a fast, accurate and robust solution for real world applications. These shortcomings are:

1. Lack of object-level consideration.

   Unlike the absolute camera pose estimation, where useful information exists throughout the entire image, objects are only visible in parts of the scene. Clearly, it is the foreground that contributes to the 6dof pose, and irrelevant background pixels should be excluded from the pipeline. This issue exists in both regression-based and reconstruction-based methods. This requirement

is more important when there are multiple interest object presented in the image.

2. Lack of uncertainty estimation in the regression-based method.

   The output from most of the pose estimation networks is deterministic rather than probabilistic, which means it is only a point estimate instead of a predictive distribution. In general, exploiting uncertainty in CNNs is an eye-catching topic, because the probability distribution of the output from CNNs can be used in a variety of ways. Most notably for our purposes, the uncertainty of the predicted pose makes the result amenable for use within the standard data fusion algorithms such as a Kalman Filter. For example in SLAM, the uncertainty of the pose estimate can be naturally fused with a state estimate to improve the accuracy of localisation over time.

3. Lack of indirect supervision in the learning process of the reconstruction-based method.

   For the reconstruction-based method, the supervisory signal for coordinates learning is crucial because it governs the mapping from image pixels to 3D points in the reference system. In some cases it maybe possible to use a pre-build 3D model of the structure (such as a map of the scene in camera relocalisation, or a 3D model of the object in object pose estimation) as the supervision, if they were available. However the existence of the fine-grained 3D model is not assured for every structure in the real world, which necessitates the investigation on indirect supervision.

Motivated by these limitations, we are hence devoted to extend the existing methods in following perspectives:

- For regression-based method, i) we aim to perform direct pose regression at the object-level, providing a **unified**, **end-to-end** framework for object pose

estimation. ii) we aim to **measure the uncertainty** of the pose prediction from regression-based camera localisation.

- For reconstruction-based method, the objective we would like to accomplish is iii) to explore **indirect supervision** for 3D coordinate learning, and iv) make it **generalize** to both camera relocalisation and object pose estimation.

Note that we investigate methods for pose estimate in end-to-end fashion (regression-based methods in chapter 3 and 4), as well as using multiple steps (reconstruction-based methods in chapter 5 and 6). It is not a problem of right or wrong when making a choice between them, because both of them have their unique advantages. For example, method using multiple steps usually achieves better accuracies because of the involved geometry, whereas end-to-end method runs very fast at the inference time, and the learnt high-level feature is pose-related which can be utilized in different reasoning tasks, *e.g.* uncertainty estimation. We will show if designed carefully and trained with sufficient data, the performance of end-to-end method could catch up or even outperform classical method with multiple steps. On top of that, when training and testing the learning-based pose estimation networks with intermediate representation (such as coordinates), the accuracy is further improved with subtle sacrifice in efficiency while geometry contributes to providing stronger constrain.

1.4 Approaches and Contribution

Throughout this thesis, we build methods for different forms of pose estimation on a learning basis, with specific solutions that address the limitations outlined in the section above:

Contribution 1 We propose a model that directly predicts the 6dof poses for objects in an image while detects and segments them from the background,

without using any 3D information at both training and inference time. It is done by isolating objects from the background and extract pose-related features only from the appearances of the objects using a CNN. We augment a novel object pose regressor to the backbone of Mask R-CNN. While detecting, segmenting and classifying the objects from the background, this regressor predicts translation and rotation parameters at the same time. The usage of Lie algebra as the representation for rotation is less-constrained, low-dimensional and less-ambiguous. For translation, an alternative 1d solution apart from the direct 3d regression method is proposed. It combines the position and size of the bounding box and the predicted z-component of the translation vector to recover the full 3d vector. The proposed end-to-end deep learning approach is able to jointly detect, segment, and directly estimate the 6dof pose of object instances from a single RGB image. The relevant work is described in chapter 3.

Contribution 2 We develop a method that combines deep learning with Gaussian Processes to produce not only a point estimate of a regressed quantity but a distribution over the value. It is achieved by combining two main machine learning frameworks, CNN and Gaussian Process Regression(GPR), formulating an end-to-end probabilistic pose inferring system. Based on the learned image feature from CNN, we use a GPR to generate the predictive distribution for the pose result. Stochastic Variational Inference is applied in our method to reduce the high complexity of GPR when dealing with large scale data, which in our case is thousands of high dimensional images. The advantage of our method is that it takes only one forward pass to estimate the uncertainty for the pose, in contrast to Bayesian PoseNet that requires multiple inferences for one test image using the dropout technique and summarizes the distribution empirically. This work is described in chapter 4.

Contribution 3 We devise a method for learning 3D scene coordinates using self-supervised learning to solve the pose estimation indirectly. This yields a method that produces state of the art camera pose estimates by combining a "best of" approach fusing learning and geometry. Specifically, we use the consistency of the 3D coordinates for the same scene point from multiple frames as self-supervision. This is done by training the coordinate regression network simultaneously with images from different viewpoints. Based on multi-view geometry, we design a loss function that contains two types of image feature reconstruction errors, along with a structural smoothness penalty over the featureless regions of the scene. Our method achieves better accuracy compared to the single-view loss methods. In addition, we observe that our method is robust to the setting of pseudo depths for different scenes, which are used to initialize the scene coordinate regression model when ground truth coordinates are missing. We show this method in chapter 5.

Contribution 4 We contribute a method that learns to regress to 3D object coordinates for object pixels without using direct supervision from 3D information of object such as a CAD model. This model is subsequently used to estimate the object pose at inference time and achieves on-par performance with the state-of-the-art object pose estimation methods, without using any 3D structural prior of the objects, which is essential to existing geometry-based methods. This proposed method follows the idea of chapter 5, building the constraints for the learning of object coordinate using multi-view geometry. To that end, a new head – the *object coordinate head* – is contributed to the Mask R-CNN backbone, whose output is the dense 3D coordinates of the object in object-centric frame. A bounding box-dependent local projection model is derived to align the pixel-to-pixel correspondence between RoI features and the object coordinate map. We also use an unsupervised learning method to

discover the equivariant features of an object, which explicitly provides the 2D-to-2D correspondences for 3D object points learning. Chapter 6 illustrates this method in detail.

Bibliography

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE, pp. 248–255.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

— (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Girshick, Ross (2015). "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*, pp. 91–99.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi (2016). "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg (2016). "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer, pp. 21–37.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Van Heel, Marin (1987). "Angular reconstitution: a posteriori assignment of projection directions for 3D reconstruction". In: *Ultramicroscopy* 21.2, pp. 111–123.

Whitaker, Ross T (1998). "A level-set approach to 3D reconstruction from range data". In: *International journal of computer vision* 29.3, pp. 203–231.

Cline, Harvey E, CL Dumoulin, HR Hart Jr, William E Lorensen, and S Ludke (1987). "3D reconstruction of the brain from magnetic resonance images using a connectivity algorithm". In: *Magnetic resonance imaging* 5.5, pp. 345–352.

Besl, Paul J and Neil D McKay (1992). "Method for registration of 3-D shapes". In: *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. International Society for Optics and Photonics, pp. 586–606.

Bajura, Michael and Ulrich Neumann (1995). "Dynamic registration correction in video-based augmented reality systems". In: *IEEE Computer Graphics and Applications* 15.5, pp. 52–60.

Blais, Gérard and Martin D. Levine (1995). "Registering multiview range data to create 3D computer objects". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.8, pp. 820–824.

Faugeras, Olivier D and Francis Lustman (1988). "Motion and structure from motion in a piecewise planar environment". In: *International Journal of Pattern Recognition and Artificial Intelligence* 2.03, pp. 485–508.

Dorfmüller, Klaus (1999). "Robust tracking for augmented reality using retroreflective markers". In: *Computers & Graphics* 23.6, pp. 795–800.

Haralick, Robert M, Hyonam Joo, Chung-Nan Lee, Xinhua Zhuang, Vinay G Vaidya, and Man Bae Kim (1989). "Pose estimation from corresponding point data". In: *IEEE Transactions on Systems, Man, and Cybernetics* 19.6, pp. 1426–1446.

Lu, Feng and Evangelos Milios (1997). "Robot pose estimation in unknown environments by matching 2d range scans". In: *Journal of Intelligent and Robotic systems* 18.3, pp. 249–275.

Dementhon, Daniel F and Larry S Davis (1995). "Model-based object pose in 25 lines of code". In: *International journal of computer vision* 15.1-2, pp. 123–141.

Covell, Michele M, Michael Hongmai Lin, Ali Rahimi, Michael Harville, Trevor J Darrell, John I Woodfill, Harlyn Baker, and Gaile G Gordon (2006). *Three dimensional object pose estimation which employs dense depth information*. US Patent 7,003,134.

Ye, Mao, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys (2011). "Accurate 3d pose estimation from a single depth image". In: *2011 International Conference on Computer Vision*. IEEE, pp. 731–738.

Hess, Wolfgang, Damon Kohler, Holger Rapp, and Daniel Andor (2016). "Real-time loop closure in 2D LIDAR SLAM". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1271–1278.

Wolcott, Ryan W and Ryan M Eustice (2014). "Visual localization within lidar maps for automated urban driving". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 176–183.

Sattler, Torsten (2013). *Efficient & effective image-based localization*. Hochschulbibliothek der Rheinisch-Westfälischen Technischen Hochschule Aachen.

Lowe, David G (1999). "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee, pp. 1150–1157.

Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool (2006). "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer, pp. 404–417.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011). "ORB: An efficient alternative to SIFT or SURF". In: *2011 International conference on computer vision*. Ieee, pp. 2564–2571.

Kendall, Alex, Matthew Grimes, and Roberto Cipolla (2015). "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946.

Kendall, Alex and Roberto Cipolla (2016). "Modelling uncertainty in deep learning for camera relocalization". In: *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, pp. 4762–4769.

— (2017). "Geometric loss functions for camera pose regression with deep learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5974–5983.

Shotton, Jamie, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon (2013). "Scene coordinate regression forests for camera relocalization in RGB-D images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937.

Brachmann, Eric, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother (2014). "Learning 6d object pose estimation using 3d object coordinates". In: *European conference on computer vision*. Springer, pp. 536–551.

Valentin, Julien, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip HS Torr (2015). "Exploiting uncertainty in regression forests for accurate camera relocalization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4400–4408.

Brachmann, Eric, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother (2017). "Dsac-differentiable ransac for camera localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6684–6692.

Brachmann, Eric and Carsten Rother (2018). "Learning less is more-6d camera localization via 3d surface regression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4654–4662.

2 Literature Review

Contents

This chapter provides a more detailed review on the publications in pose estimation. We start by reviewing the work that is proposed before the era of deep learning, which mostly rely on detecting sparse features from image. Then we address the limitations of hand-crafted features. In the remaining part of this chapter, the modern methods that use deep learning for pose estimation are reviewed. Though two tasks are focused in this thesis, camera relocalisation and object pose estimation, the main ideas behind the solutions to them share large similarities. Therefore we use camera relocalisation as an example to introduce related work in this chapter, and

review the publications specific to object pose estimation in corresponding chapters

2.1 Related Work Before Era of CNNs

The objective of relocalisation is to estimate the global 6dof pose of the camera/robot in a known scene. The most famous problem that relates to global localisation is the kidnapped robot problem proposed by Se et al. 2001, in which a robot is 'kidnapped' and moved to an unknown location. The robot is expected to recall the pose for the unknown location using information from its memory, which is represented by a database of image-pose pairs. Another typical application of camera relocalisation is loop closure in visual Simultaneous Localisation and Mapping SLAM (Newman et al. 2005; Davison et al. 2007; Clemente et al. 2007; Ho et al. 2007; Labbe et al. 2013; Ho et al. 2006). Building on the idea of relocalisation, loop closure detector of a full SLAM algorithm recognizes the occurrence of previously visited location in the online-built map for robots and corrects the pose drift caused by the frame-to-frame motion tracker (*i.e.* visual odometry).

As mentioned before, there are mainly two ways of solving camera relocalisation in previous work: (i) retrieval-based approach and (ii) correspondences-based approach.

2.1.1 Retrieval-based Methods.

The general idea of retrieval-based methods is to extract correlated information from the pre-built database for a query image by matching a closest training images according to a measurement of sensory similarity. The pose for the query image is further determined based on the pose of the closest match (or matches).

From a complexity perspective, an image itself is redundant and high-dimensional. Using the raw image as the representation for image-matching apparently is a poor choice in terms of speed. In order to improve efficiency, Ulrich et al. 2000 represent the appearance as a collection of image histograms, and Lamon et al. 2001 summarize

the images using ordered lists of edges and intensity features. However, these methods only find a topological pose for the query image, rather than estimating a new pose that has 6dof, which apparently needs a further geometry verification.

Given a collection of image-pose pairs of a building created from a hand-set, Robertson et al. 2004 assume a canonical view for the facade and rectify the images to detect viewpoint-invariant features, resulting in an offline database. Wide baseline matching is carried out for the query image by matching feature to the database. Another work that performs wide baseline matching for global localisation is done by W. Zhang et al. 2006. They alternatively use viewpoint-invariant SIFT feature (Lowe 2004) to describe image keypoints. They then compute the position relative to the two most similar training images and use known GPS coordinates of these images to achieve the absolute position. These two methods match the reference frame to the training images by conducting the matching between individual features, instead of use the information of two images as a single metric. The former is time-consuming when the size of the database is large.

Inspired by the idea of Google text search by Baeza-Yates et al. 1999, Sivic et al. 2003 propose to simulate the behavior of the text retrieval for images. They create a vector of words for an image that counts the occurrence of SIFT feature in it. These feature descriptors are vector quantized into clusters which will be the visual vocabulary and referred as Bag-of-Words (BoWs). It enables the comparisons with thousands of images happen in dozens of milliseconds (Nister et al. 2006).

Following this technique, Cummins et al. 2008 extract and vectorise SURF features (Bay et al. 2006) from the training images as visual vocabulary. The query image is also abstracted in the same way and matched with the database probabilistically using a graph model based on the co-occurrences of certain features. To further speed up the feature extraction, Galvez-Lopez et al. 2011 use FAST (Rosten et al. 2006) key-points and BRIEF (Calonder et al. 2010) descriptors to build the BoWs for image matching and Gálvez-López et al. 2012

build a tree structure for vocabulary that discretizes a binary descriptor space and use it to speed up image-retrieval.

2.1.2 Correspondences-based Methods.

The retrieval-based methods show advantages in terms of speed for global localisation, whereas correspondences-based methods naturally build stronger geometric constraints for 6dof pose by matching 2D images points and 3D coordinates.

Unlike retrieval-based methods that use the a collection of raw images as the representation for a scene, a 3D map is usually created for correspondences-based methods to perform 2D-3D points matching. This 3D map, which contains visual landmarks and their 3D coordinates in the world space, can be obtained online or offline, depending on the availability of the scene prior. If robot visits an unknown environment, the map of the scene can be built incrementally from the acquired image sequence from camera using visual SLAM. On the other hand, when the training images of a scene are given and the objective is to estimate a new pose in the scene for a query that contains similar landmarks, the 3D map or the model of the scene can be reconstructed from images by parallel tracking and mapping (PTAM) (Klein et al. 2007; Castle et al. 2008; Klein et al. 2009) or Structure-from-Motion(SfM) (Agarwal et al. 2011; Snavely et al. 2008; Wu 2013; Schonberger et al. 2016; Snavely et al. 2006).

For the representation of landmarks in the map, work have used pre-defined artificial landmarks such as barcode (Everett et al. 1995) or QR code (H. Zhang et al. 2015) for fast and easy recognition. In a general environment, Sim et al. 1999 extract the features of landmarks from image based on visual attention. Interest points are tracked in multiple views and parameterized with a set of attributes, such as position in the image, intensity distribution, edge distribution, *etc.* In order to improve the degree of viewpoint-independence, the famous SLAM system by Davison 2003 and Williams et al. 2007a detects Shi-Tomasi corner (Shi et al.

1994) in a relatively large image patch and discover landmarks within them. SIFT feature (Lowe 1999) enhances this invariance and therefore is widely used in the landmark characterization for correspondence-based methods, such as (Se et al. 2002; Karlsson et al. 2005; Irschara et al. 2009). Recently, ORB features by Rublee et al. 2011 improves the efficiency of SIFT which requires relatively more extensive computing power in real time.

In the subsequent pose estimation stage, Se et al. 2001 matches a set of SIFT features of the query image to the database, and applies Hough Transform (Hough 1962) on a discrete pose space to find the pose that produce the maximum feature-landmark correspondences. Work in object pose recognition (Lepetit et al. 2006; Özuysal et al. 2006) trains a forest of fast classifiers using the image patches in the database offline. The trained forest then classifies the query features to establish correspondences with the database. Leveraging from this idea, Williams et al. 2007b deploy a similar randomised lists key-point recognition algorithm, training the forest online with the features from mapping of SLAM. When a new frame comes, its features firstly are classified by this tree, yielding correspondences which can be used to relocalise the new image. Based on the pose verification, this method prevents the system from failure of pose tracking.

For the offline 3D models created from SfM of a very large scene, Irschara et al. 2009 conduct a two-step scheme for correspondence establishment. They firstly retrieve the similar image for the query image in the database, which contains potential matches for the features. Then direct match is performed for the query features and the retrieved segment of the whole model. In order to directly and efficiently match 2D feature to 3D points in the database, Sattler et al. 2011 associates 3D points with a visual vocabulary obtained from clustering SIFT features. The linear search space for each query descriptor is prioritized based on the number of descriptors assigned to its corresponding visual word. In work proposed by Sattler et al. 2012, the authors consider to actively search correspondences

in both 2D-to-3D and 3D-to-2D direction, making the registration possible for a million 3D points in real time.

2.1.3 Limitation of Feature-based Methods

Although perform efficiently in detection and matching, these human-designed feature extractors unfortunately have several limitations:

- Sparsity. Only a small set of key-points are considered as candidates for feature extraction. Large amount of visual information of the world is ignored by sparsification, which reduces the number of potential correspondences and subsequently jeopardizes the robustness of the pose calculation;

- Limited scales of receptive field. The feature only describes the characteristic of a window around each key-point hence is bereft of global (or intermediate-level) information;

- Despite the success of handcraft features like SIFT, the advent of CNNs show these features were not achieving as good a compromise between distinctiveness and invariance. The features extracted from a CNN have been shown to be better descriptors that balance between distinctiveness and invariance.

- Poor scalability. The numbers of features increases while the scene scales up. It is not considered as an issue in a small scale of scene or the case of object pose estimation, however for a large outdoor scene, the space for storage and complexity for matching grows rapidly, which is disadvantageous for real time application.

In order to overcome these shortcoming, researchers propose to introduce deep learning, most notably CNNs to the task of pose estimation.

2.2 Neural Networks

2.2.1 Artificial Neural Networks

Inspired by biological neural networks, artificial neural networks have been investigated to simulate the behavior of how information transfers in the recognition system of human being. From a high level, neural networks can be viewed as a non-linear model that maps the input to an output space, but what makes them different from other learning models (such as Support Vector Machine and Random Forest, *etc.*) is that the structure of a neural network consists of thousands of neurons, which are grouped into different layers to pass and process the information from input end to output end. These layers are defined as hidden layers and fully connected with their neighbor hidden layers. This pattern forms the structure of network. Activation functions are applied to the hidden layer to produce the output. When learning the network, a loss function compares the output and ground truth data to build an objective to optimize. Since the nodes are fully connected between hidden layers, the dimensionality of network scales exponentially with the size of input, which yields demanding resource for training.

2.2.2 Convolutional Neural Networks

LeCun et al. 1998 propose to use convolutional layers to replace fully connected hidden layer as the building block of a network to take high dimensional input such as image. It also consists of pooling layers, activation function and loss function. See figure 2.1 for the architecture of LeNet-5.

Convolution Layer: Convolution layers are commonly represented by kernels, which consist of a set of weights that are shared across the input. The small spatial size (such as 3×3 or 5×5 and known as *receptive field*) of such kernel reduces the complexity of the network. During a forward pass, kernels are convolved across the spatial dimensions of the input, computing the dot product between the weights

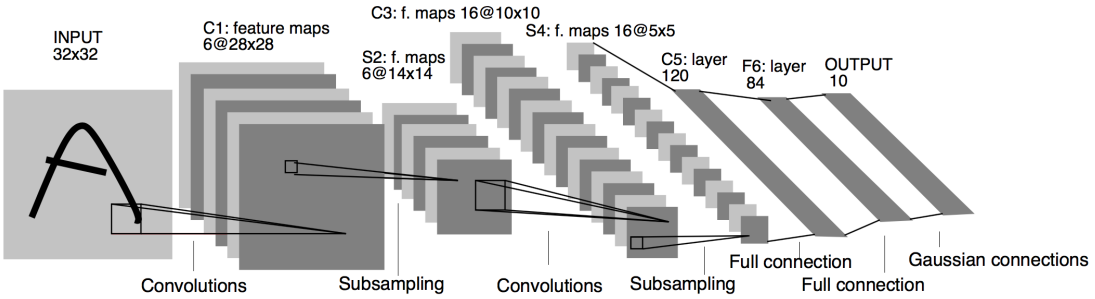


Figure 2.1: The architecture of LeNet-5 (LeCun et al. 1998) for hand written character recognition. Figure is from (LeCun et al. 1998)

and each connected block of the input. The kernels of convolution layers can be viewed as feature detectors, which means that the network learns to activate when they find specific pattern inside the local region. In this way, the individual element of the output feature map or activation map is locally connected with its input. With stacked convolution layers, the receptive field of feature at the output end grows to recognize the global information of the input.

To further save computation, a larger stride can be applied to convolution layers when sliding the kernel across the input, which makes it as a sub-sampler to reduce the spatial dimensions of activation maps.

Pooling Layer. As an alternative way of decreasing feature size and increasing receptive field, pooling layers perform sub-sampling using max or average operation on feature maps. Max pooling selects the strongest activation from a local region of the feature map and ignores the rest, whereas average pooling averages the responses within the window. In this way, the invariance to local spatial particularity is introduced to the network by pooling layers .

Activation Layer. In order to increase non-linearity of CNNs, activation functions are applied to the output feature map of convolution layers. Sigmoid activation layer was proposed to play the role of non-linear activation, but it may cause vanishing gradient during training because of neuron saturation. Nair et al. 2010

introduce Rectified Linear Units (ReLU) to overcome this problem. A ReLU essentially preserves the input when it is positive and outputs zero otherwise. Its function is $ReLU(x) = max(0, x)$. ReLU has become the most commonly used activation function because of efficiency in training and generalization ability (Krizhevsky et al. 2012).

Fully Connected Layer. In order to reason from high-level information of the input learnt by CNNs, the activation map from the last convolution layer is flattened to a vector and forwarded to fully connected layer. It builds the ultimate connection between input and output by performing dot multiplication. It is also able to transfer information between modalities when placed before the final output layer.

Other Layers. Dropout layer (Srivastava et al. 2014) randomly set some neurons of the activation map to zeros during training to prevent the complex CNN from overfitting. In addition, Batch Normalization Layer proposed by Ioffe et al. 2015 can also avoid overfitting and speed up the training by normalizing the input of a layer within a training batch.

2.3 Deep Learning in Pose Estimation

CNNs have been applied to the problem of pose estimation recently to overcome the limitation addressed in section 2.1.3. As mentioned in section 1.2, we categorise them into two groups according to their relationship to the classical methods, namely regression-based method and reconstruction-based method.

2.3.1 Regression-based Learning

Firstly, the query image is 'retrieved' against database in image-level using global, high-level features from CNNs. Such image features are learnt by training a neural network to map the input images to their poses. These CNNs commonly consist of

two parts: a feature extractor and a pose regressor. As investigated in (Kendall et al. 2015), the high-dimensional image feature vectors from the CNN extractor trained with pose information are strongly related to the positions of landmarks such as windows and spires, from which the camera pose of an image can be determined. A pose regressor is also trained at the same time to transfer these features to the ground truth poses of training images. At inference time, pose for query image is essentially interpolated by the training poses using the inferred image feature as the input to the regressor. This line of work (Kendall et al. 2015; Kendall et al. 2016; Kendall et al. 2017; Brahmbhatt et al. 2018; Melekhov et al. 2017; Naseer et al. 2017; Walch et al. 2017; Henriques et al. 2018) is also known as Absolute Pose Regression (APR),

More specifically, PoseNet introduced by Kendall et al. 2015 pioneers the idea of applying a deep learning model to the problem of camera pose estimation. A CNN is used to extract high-dimensional features directly from the RGB image, followed by two fully connected layers to regress the translation vector and the rotation quaternion. Although the model is robust to dynamic changes of the scene due to its high-level generalization ability, the performance of PoseNet (Kendall et al. 2015) and its variants (Kendall et al. 2016; Kendall et al. 2017; Walch et al. 2017) do not perform sufficiently well for accurate localisation. Nevertheless, the most notable improvement comes from the subsequent geometric loss based PoseNet (Kendall et al. 2017). It leverages the physical model of the scene and supervises the learning of the pose regression model by minimizing the reprojection error of the 3D points, eliminating the dependence on the choice of hyperparameters between translational and rotational losses. Moreover, a homoscedastic task loss is also used to learn the model, which relies on RGB information only and achieves on-par performance to the RGB-D version. The need of the 3D model however means that this method is inapplicable when only RGB images are at hand. Recently, Balntas et al. 2018 proposed RelocNet that relies on evaluating the similarity between the query image and images in the training database. The pose of the query image is

then recovered based on the absolute pose of its nearest neighbor and the estimated relative transformation between them. Despite the progress, there is much room for improvement in accuracy for these methods.

2.3.2 Reconstruction-based Learning

In an alternative approach, the points of a new image are 'matched' against a 3D structure, by performing coordinate regression using deep model at the pixel(or point)-level. Matching is obtained by designing a CNN, which regresses to a set of 3 values for every pixel, where the 3 values are the 3D coordinates in an absolute coordinate frame. For instance, this coordinate frame is world-centric if the task is camera relocalisation, whereas it is object-centric for object pose estimation. Representatives are (Shotton et al. 2013; Brachmann et al. 2014; Valentin et al. 2015; Brachmann et al. 2017; Brachmann et al. 2018). The learning of these methods also can be considered as an implicit process of 3D reconstruction of the target structure from all images in the database. At inference time, this CNN acts as an establisher for correspondence between images pixels and 3D points in the reference frame. Pose is then solved according to these correspondence using classical geometric algorithm, *i.e.* Perspective-n-Point (PnP) solvers (Gao et al. 2003; Lepetit et al. 2009; Hesch et al. 2011; Wang et al. 2018). We name them as *reconstruction-based* method in the following.

The idea of using scene coordinates to obtain dense 2D-3D correspondences is initially proposed by Shotton et al. 2013. A Random Forest is trained to infer the 3D scene (world) coordinate for image pixels from RGB-D data. The RANSAC (Fischler et al. 1981) pipeline is then revisited to estimate the camera pose accurately. Valentin et al. 2015 exploits the uncertainty in the estimate from the Random Forest to benefit the pose optimization.

DSAC (Brachmann et al. 2017) and DSAC++ (Brachmann et al. 2018) deploy two versions of an end-to-end scene coordinate regressor based on CNNs, and are

devoted to make all the steps in the traditional RANSAC differentiable to enable an end-to-end training pipeline. In DSAC (Brachmann et al. 2017), the CNN for scene coordinate regression takes a small patch of the image as the input, and its output is the 3D coordinate associated to the central pixel of the input patch. As an ameliorator, DSAC++ (Brachmann et al. 2018) was upgraded to a fully convolutional network (FCN) (Long et al. 2015) to improve the efficiency of training and to preserve the image-patch-to-coordinate property. To perform the three-step RANSAC algorithm, they start by sampling a pool of pose hypotheses using the PnP solver over the dense 2D-3D correspondences given by the scene coordinate prediction. In the second stage of ranking the hypotheses, DSAC (Brachmann et al. 2017) scores them with another CNN whose input is the reprojection error map of the predicted scene coordinates given each pose hypothesis and the camera intrinsics. On the other hand, to overcome the overfitting issue of the scoring CNN in DSAC (Brachmann et al. 2017), DSAC++ (Brachmann et al. 2018) simply uses a soft inliers counting scheme to evaluate the merits of the hypotheses. The difference also exists in the last refinement step. To make this iterative procedure differentiable, DSAC (Brachmann et al. 2017) approximates the gradient via finite differences, and DSAC++ (Brachmann et al. 2018) uses the iterative Gauss-Newton algorithm to linearise the model. Combining these techniques, they achieve the state-of-the-art result for camera relocalisation in both indoor and outdoor scenes, even without the 3D model of scene.

Bui et al. 2018 also estimate the confidence/uncertainty of the scene coordinates as an auxiliary prediction from the network, and then run RANSAC using those inferred coordinates that have high confidence, which improves the robustness of the system.

2.4 Summary

In this chapter, we have reviewed the literatures proposed for the task of pose estimation, covering methods developed before and after the era of deep learning. The classical methods are of two main categories: retrieval-based methods and correspondences-based methods. Regardless of differences between them, sparse artificial features computed for image keypoints are used in their formulation and therefore limit their applications in real world environment. Learning is introduced into modern solutions to address these limitations. Promising accuracy has been achieved, while robustness to environment dynamics being guaranteed to some extent. In the family of regression-based learning and reconstruction-based learning approaches, the 3D CAD models are deeply involved in their solutions, providing the structural prior of the scenes or objects, which help algorithms to utilize the geometry to conduct accurate pose estimation. With the next chapter being a starter, we introduce our proposed learning-based methods for 6dof pose estimation, which do not rely on the 3D CAD models and sacrifice no performance in terms of accuracy.

Bibliography

Se, Stephen, David Lowe, and Jim Little (2001). "Local and global localization for mobile robots using visual landmarks". In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. Vol. 1. IEEE, pp. 414–420.

Newman, Paul and Kin Ho (2005). "SLAM-loop closing with visually salient features". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, pp. 635–642.

Davison, Andrew J, Ian D Reid, Nicholas D Molton, and Olivier Stasse (2007). "MonoSLAM: Real-time single camera SLAM". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6, pp. 1052–1067.

Clemente, Laura A, Andrew J Davison, Ian D Reid, José Neira, and Juan D Tardós (2007). "Mapping Large Loops with a Single Hand-Held Camera." In: *Robotics: Science and Systems*. Vol. 2. 2.

Ho, Kin Leong and Paul Newman (2007). "Detecting loop closure with scene sequences". In: *International Journal of Computer Vision* 74.3, pp. 261–286.

Labbe, Mathieu and Francois Michaud (2013). "Appearance-based loop closure detection for online large-scale and long-term operation". In: *IEEE Transactions on Robotics* 29.3, pp. 734–745.

Ho, Kin Leong and Paul Newman (2006). "Loop closure detection in SLAM by combining visual and spatial appearance". In: *Robotics and Autonomous Systems* 54.9, pp. 740–749.

Ulrich, Iwan and Illah Nourbakhsh (2000). "Appearance-based place recognition for topological localization". In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 2. Ieee, pp. 1023–1029.

Lamon, Pierre, Illah Nourbakhsh, Björn Jensen, and Roland Siegwart (2001). "Deriving and matching image fingerprint sequences for mobile robot localization". In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*. Vol. 2. IEEE, pp. 1609–1614.

Robertson, Duncan P and Roberto Cipolla (2004). "An Image-Based System for Urban Navigation." In: *Bmvc*. Vol. 19. 51. Citeseer, p. 165.

Zhang, Wei and Jana Kosecka (2006). "Image Based Localization in Urban Environments." In: *3DPVT*. Vol. 6. Citeseer, pp. 33–40.

Lowe, David G (2004). "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2, pp. 91–110.

Baeza-Yates, Ricardo, Berthier Ribeiro-Neto, et al. (1999). *Modern information retrieval*. Vol. 463. ACM press New York.

Sivic, Josef and Andrew Zisserman (2003). "Video Google: A text retrieval approach to object matching in videos". In: *Proceedings of International Conference on Computer Vision*. IEEE, p. 1470.

Nister, David and Henrik Stewenius (2006). "Scalable recognition with a vocabulary tree". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. Ieee, pp. 2161–2168.

Cummins, Mark and Paul Newman (2008). "FAB-MAP: Probabilistic localization and mapping in the space of appearance". In: *The International Journal of Robotics Research* 27.6, pp. 647–665.

Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool (2006). "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer, pp. 404–417.

Galvez-Lopez, Dorian and Juan D Tardos (2011). "Real-time loop detection with bags of binary words". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 51–58.

Rosten, Edward and Tom Drummond (2006). "Machine learning for high-speed corner detection". In: *European conference on computer vision*. Springer, pp. 430–443.

Calonder, Michael, Vincent Lepetit, Christoph Strecha, and Pascal Fua (2010). "Brief: Binary robust independent elementary features". In: *European conference on computer vision*. Springer, pp. 778–792.

Gálvez-López, Dorian and Juan D Tardos (2012). "Bags of binary words for fast place recognition in image sequences". In: *IEEE Transactions on Robotics* 28.5, pp. 1188–1197.

Klein, Georg and David Murray (2007). "Parallel tracking and mapping for small AR workspaces". In: *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, pp. 1–10.

Castle, Robert, Georg Klein, and David W Murray (2008). "Video-rate localization in multiple maps for wearable augmented reality". In: *2008 12th IEEE International Symposium on Wearable Computers*. IEEE, pp. 15–22.

Klein, Georg and David Murray (2009). "Parallel tracking and mapping on a camera phone". In: *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, pp. 83–86.

Agarwal, Sameer, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski (2011). "Building rome in a day". In: *Communications of the ACM* 54.10, pp. 105–112.

Snavely, Noah, Steven M Seitz, and Richard Szeliski (2008). "Modeling the world from internet photo collections". In: *International journal of computer vision* 80.2, pp. 189–210.

Wu, Changchang (2013). "Towards linear-time incremental structure from motion". In: *2013 International Conference on 3D Vision-3DV 2013*. IEEE, pp. 127–134.

Schonberger, Johannes L and Jan-Michael Frahm (2016). "Structure-from-motion revisited". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113.

Snavely, Noah, Steven M Seitz, and Richard Szeliski (2006). "Photo tourism: exploring photo collections in 3D". In: *ACM transactions on graphics (TOG)*. Vol. 25. 3. ACM, pp. 835–846.

Everett, Hobart R, Douglas W Gage, Gary A Gilbreath, Robin T Laird, and Richard P Smurlo (1995). "Real-world issues in warehouse navigation". In: *Mobile Robots IX*. Vol. 2352. International Society for Optics and Photonics, pp. 249–259.

Zhang, Huijuan, Chengning Zhang, Wei Yang, and Chin-Yin Chen (2015). "Localization and navigation using QR code for mobile robot in indoor environment". In: *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, pp. 2501–2506.

Sim, Robert and Gregory Dudek (1999). "Learning and evaluating visual features for pose estimation". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. IEEE, pp. 1217–1222.

Davison, Andrew J (2003). "Real-time simultaneous localisation and mapping with a single camera". In: *Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE, p. 1403.

Williams, Brian, Paul Smith, and Ian Reid (2007a). "Automatic relocalisation for a single-camera simultaneous localisation and mapping system". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, pp. 2784–2790.

Shi, Jianbo et al. (1994). "Good features to track". In: *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE, pp. 593–600.

Lowe, David G (1999). "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee, pp. 1150–1157.

Se, Stephen, David Lowe, and Jim Little (2002). "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks". In: *The international Journal of robotics Research* 21.8, pp. 735–758.

Karlsson, Niklas, Enrico Di Bernardo, Jim Ostrowski, Luis Goncalves, Paolo Pirjanian, and Mario E Munich (2005). "The vSLAM algorithm for robust localization and mapping". In: *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, pp. 24–29.

Irschara, Arnold, Christopher Zach, Jan-Michael Frahm, and Horst Bischof (2009). "From structure-from-motion point clouds to fast location recognition". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2599–2606.

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011). "ORB: An efficient alternative to SIFT or SURF". In: *2011 International conference on computer vision*. Ieee, pp. 2564–2571.

Hough, Paul VC (1962). *Method and means for recognizing complex patterns*. US Patent 3,069,654.

Lepetit, Vincent and Pascal Fua (2006). "Keypoint recognition using randomized trees". In: *IEEE transactions on pattern analysis and machine intelligence* 28.9, pp. 1465–1479.

Özuysal, Mustafa, Vincent Lepetit, François Fleuret, and Pascal Fua (2006). "Feature harvesting for tracking-by-detection". In: *European conference on computer vision*. Springer, pp. 592–605.

Williams, Brian, Georg Klein, and Ian Reid (2007b). "Real-time SLAM relocalisation". In: *2007 IEEE 11th international conference on computer vision*. IEEE, pp. 1–8.

Sattler, Torsten, Bastian Leibe, and Leif Kobbelt (2011). "Fast image-based localization using direct 2d-to-3d matching". In: *2011 International Conference on Computer Vision*. IEEE, pp. 667–674.

— (2012). "Improving image-based localization by active correspondence search". In: *European conference on computer vision*. Springer, pp. 752–765.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167*.

Kendall, Alex, Matthew Grimes, and Roberto Cipolla (2015). "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946.

Kendall, Alex and Roberto Cipolla (2016). "Modelling uncertainty in deep learning for camera relocalization". In: *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, pp. 4762–4769.

— (2017). "Geometric loss functions for camera pose regression with deep learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5974–5983.

Brahmbhatt, Samarth, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz (2018). "Geometry-aware learning of maps for camera localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625.

Melekhov, Iaroslav, Juha Ylioinas, Juho Kannala, and Esa Rahtu (2017). "Image-based localization using hourglass networks". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 879–886.

Naseer, Tayyab and Wolfram Burgard (2017). "Deep regression for monocular camera-based 6-dof global localization in outdoor environments". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1525–1530.

Walch, Florian, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers (2017). "Image-based localization using lstms for structured feature correlation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 627–637.

Henriques, Joao F and Andrea Vedaldi (2018). "Mapnet: An allocentric spatial memory for mapping environments". In: *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8476–8484.

Balntas, Vassileios, Shuda Li, and Victor Prisacariu (2018). "Relocnet: Continuous metric learning relocalisation using neural nets". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 751–767.

Shotton, Jamie, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon (2013). "Scene coordinate regression forests for camera

relocalization in RGB-D images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937.

Brachmann, Eric, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother (2014). "Learning 6d object pose estimation using 3d object coordinates". In: *European conference on computer vision*. Springer, pp. 536–551.

Valentin, Julien, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip HS Torr (2015). "Exploiting uncertainty in regression forests for accurate camera relocalization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4400–4408.

Brachmann, Eric, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother (2017). "Dsac-differentiable ransac for camera localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6684–6692.

Brachmann, Eric and Carsten Rother (2018). "Learning less is more-6d camera localization via 3d surface regression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4654–4662.

Gao, Xiao-Shan, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng (2003). "Complete solution classification for the perspective-three-point problem". In: *IEEE transactions on pattern analysis and machine intelligence* 25.8, pp. 930–943.

Lepetit, Vincent, Francesc Moreno-Noguer, and Pascal Fua (2009). "Epnp: An accurate o (n) solution to the pnp problem". In: *International journal of computer vision* 81.2, p. 155.

Hesch, Joel A and Stergios I Roumeliotis (2011). "A direct least-squares (DLS) method for PnP". In: *2011 International Conference on Computer Vision*. IEEE, pp. 383–390.

Wang, Ping, Guili Xu, Zhengsheng Wang, and Yuehua Cheng (2018). "An efficient solution to the perspective-three-point pose problem". In: *Computer Vision and Image Understanding* 166, pp. 81–87.

Fischler, Martin A and Robert C Bolles (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6, pp. 381–395.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Bui, Mai, Shadi Albarqouni, Slobodan Ilic, and Nassir Navab (2018). "Scene coordinate and correspondence learning for image-based localization". In: *arXiv preprint arXiv:1805.08443*.

3

An End-to-End Learning-based Method for Direct Object Pose Estimation

Contents

We introduce our direct regression-based object pose estimation method in this chapter. Our proposed method detects and segments object instances in the image

using Mask R-CNN, and contains a novel additional pose head for 6D pose estimation. This head estimates the rotation matrix of an object by regressing a Lie algebra based rotation representation, and estimates the translation vector by predicting the distance of the object to the camera center. This work has been published to British Machine Vision Conference 2018. As the third author[1] *in the list, my contribution involved the design of the the initial problem with the first author and my supervisor, as well as the design and implementation of the pose regression branch. I also worked on data preparation for training and test.*

3.1 Introduction

In this chapter, we aim at finding a solution to the first limitation of learning-based pose estimation that we mentioned in section 1.3, proposing a method that performs direct 6dof pose estimation for known object instances in an end-to-end fashion, from only a single RGB input.

Similar to the idea of correspondence-based camera relocalisation, traditional object pose estimation methods such as (Gordon et al. 2006; Martinez et al. 2010; Wagner et al. 2008) are mainly based on matching hand-crafted local features (Lowe 2004) in 2D image and 3D object model. However, local feature matching approach is only suitable for objects with rich textures. Template-based matching methods (Hinterstoisser et al. 2012; Rios-Cabrera et al. 2013) are used for object with poor texture. Unfortunately, they are usually sensitive to light changes and partial occlusions. Although state-of-the-art deep feature learning approaches (Brachmann et al. 2014; Krull et al. 2015; Michel et al. 2017) can provide more accurate pose than the template approaches, they are heavily dependent on depth. This structural information is used for both generating and selecting pose

[1] At the time of development and publication, there was no open source implementation of Mask R-CNN, so the first author of the paper created one and that constituted a major development effort (which was also a reason why he was first author on the paper).

hypotheses steps. As an extra helper, the 3D models of the objects are also needed in a subsequent pose refinement process.

Different from these learning-based 6dof pose estimation methods which rely on RGB-D inputs, in this chapter, we develop a deep neural network which can recover the 6dof poses of object instances from pure RGB information. It is done by utilizing the recently proposed advanced region-based CNNs (Girshick 2015; Ren et al. 2015), based on which we directly regress the 6dof pose for each object instance in a single forward pass, without using any knowledge of the depth or the 3D models of the interest objects.

Object classification (Krizhevsky et al. 2012), detection (Girshick 2015; Ren et al. 2015), and recent instance segmentation (He et al. 2017) have achieved huge improvements using CNNs. However, the application of CNNs to 6dof object pose estimation problem is still limited (at the time that this chapter was published). There are a few regression-based work which use CNNs for direct 6dof camera pose regression (Kendall et al. 2015; Kendall et al. 2017). However, compared to camera pose estimation (which consider the whole world as an object), object pose estimation requires to detect, segment, and recover the pose for every object instance in the image.

The key for recent achievement in object detection and object instance segmentation is the development of a region-based CNN, *i.e.* a Region Proposal Network (RPN) by Girshick 2015. RPN is fundamentally a CNN which is trained to produce multiple object bounding box proposals in an image at different shapes and sizes. Faster R-CNN by Ren et al. 2015 further refines bounding boxes produced by RPN and simultaneously classifies bounding box labels in a single forward pass. The recent work Mask R-CNN (He et al. 2017) goes beyond detection, performing binary segmentation in each bounding box from RPN. Despite its simple design, Mask R-CNN achieves state-of-the-art results for instance segmentation.

Note that in both Faster R-CNN and Mask R-CNN, the training and testing of the network are both in an end-to-end fashion. For example, Mask R-CNN simultaneously localizes, classifies, and segments object instance, outputting these three predictions in a single forward pass. Leveraging the impressive results of Mask R-CNN for object detection and instance segmentation – which are two key components in a 6dof pose estimation problem – we are motivated to find the answer for the question that, *can we exploit the merits of RPN to recover the 6dof poses of object instances from a single RGB image in an end-to-end fashion?*

To this end, we design a network which extends Mask R-CNN by adding a new branch for regressing the pose of the objects inside bounding boxes produced by RPN. The contributed pose branch is in parallel with the existing branches for bounding box recognition and instance segmentation. It predicts poses by estimating translation and rotation parameters separately. The translation of an object is estimated by combining the position of the bounding box in the image (given by the bounding box branch) and the distance of the object center to the camera (given by the pose branch). Care must be taken when regressing the rotation matrix as not all 3×3 matrices are valid rotation matrices. We choose to use Lie algebra as the representation for rotation. The Lie algebra of the rotation matrix group parameterises a rotation with only three scalar values. Such representation is unconstrained and not over-parameterised, thus well suited for regression with deep learning.

As a result, our proposed model is simple and elegant, and it does not require an expensive pose refinement post-process. It allows fast inference at about 100ms per frame on a GPU. Evaluated on two standard pose benchmarking datasets, our method surpasses all the state-of-the-art RGB pose estimation methods that are used without post-refinements.

3.2 Related Work

In this section, we firstly complete the literature review in chapter 2, focusing on the recent 6dof object pose estimation work and the application of CNNs for 6dof pose problems. We then briefly cover the main design of the recent methods which are based on RPN for object detection and segmentation.

3.2.1 Object 6dof Pose Estimation

The topic of pose estimation has widely studied in the literature. For objects with rich textures, sparse feature matching approaches have shown good performance in terms of accuracy (Gordon et al. 2006; Lowe 2004). They are however not reliable and robust to texture-less objects. For this non-trivial situation, PWP3D by Prisacariu et al. 2012 relies on level-set methods to maximize the discrimination between the statistics of foreground and background of object. It achieves both segmentation and 6dof pose at the same time by making the level set energy function differentiable with respect to the pose parameters. Hinterstoisser et al. 2012 and Tejani et al. 2014 propose to use object template to overcome the lack of texture. The most notable work belonging to this category is LINEMOD (Hinterstoisser et al. 2012) which uses stable gradient and normal features for template matching. However, LINEMOD is designed to work with RGB-D images. Furthermore, template-based approaches are sensitive to illuminations and occlusions.

Recent 6dof pose estimation researches have relied on feature learning to deal with objects that have insufficient texture (Brachmann et al. 2014; Krull et al. 2015; Michel et al. 2017). In (Brachmann et al. 2014; Krull et al. 2015), the authors show that the dense feature matching methods outperform traditional approaches. The basis design of (Brachmann et al. 2014; Krull et al. 2015; Michel et al. 2017) is a multi-stage scheme, i.e., a random forest is used for jointly learning the object category for pixels in the image (known as object labels) and the coordinate of the

pixels w.r.t. object coordinate systems (known as object coordinates). A set of pose hypotheses are generated by using the outputs of the forest and the depth channel of the input image. RANSAC is then performed to obtain a pose that has maximum agreement between all matches.

However, the pipelines in (Brachmann et al. 2014; Krull et al. 2015) depend heavily on the depth channel. The depth information is required in both pose hypothesis generation and refinements. The work by Brachmann et al. 2016 also follows a multi-stage approach as (Brachmann et al. 2014; Krull et al. 2015) but is designed to work with RGB images. They use auto-context random forest to improve the prediction of the object labels and the object coordinates. In order to deal with the missing depth information, the distribution of object coordinates is approximated as a mixture model and used when generating pose hypothesis.

One of drawbacks of feature learning approaches (Brachmann et al. 2014; Krull et al. 2015; Brachmann et al. 2016) is that the generation of pose hypotheses uses only local information, i.e., only three or four pixels are used to generate a hypothesis. As result, this may generate bad hypotheses because it does not consider a global context over the whole object. Furthermore, by requiring multiple processing steps, the learning approaches in (Brachmann et al. 2014; Krull et al. 2015; Brachmann et al. 2016) are time-consuming, making them unsuitable for real-time applications. In contrast to most aforementioned approaches, we recover object pose from a single RGB image. In addition, instead of generating pose hypotheses by using only local information and refine them as the previous works, we rely on global information, i.e., whole object, to directly regress the pose.

Recently, CNN has been applied for 6dof pose problem, but it is mostly for camera pose, *e.g.* (Kendall et al. 2015; Kendall et al. 2017). Camera pose estimation and object pose estimation are pretty-much dual (or same) problems, except that object pose is harder to compute than camera pose, due to the fact that objects are always a subset of the image, which requires an additional step of detection.

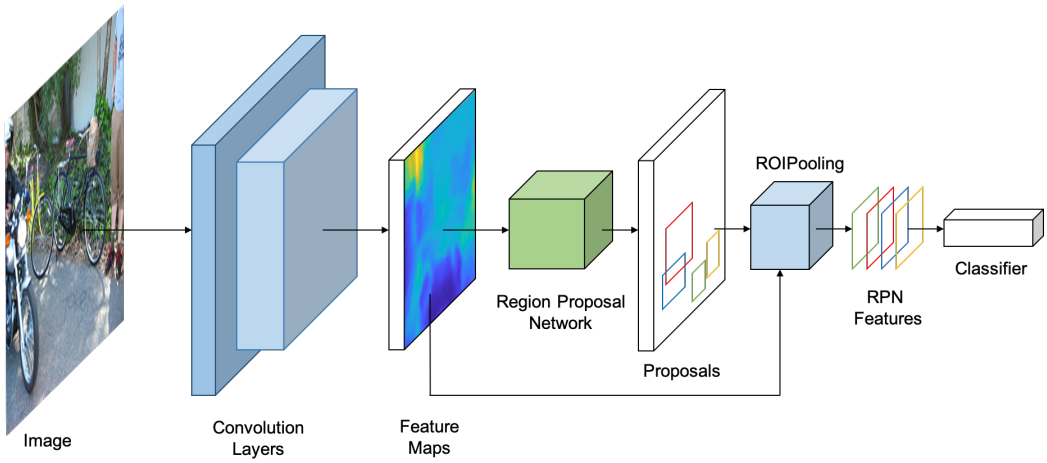


Figure 3.1: Architecture of Faster R-CNN. Recreated from (Ren et al. 2015).

Brachmann et al. 2016 apply a CNN into their object pose estimation system. However, in that work, the CNN acts as a probabilistic model to learn to compare the learned information (produced by a random forest) and the rendered image. In contrast to (Brachmann et al. 2016), we use CNN as a regressor which directly regresses object poses from a single RGB input.

Instead of directly regressing object poses, other recent methods (Rad et al. 2017; Kehl et al. 2017; Tekin et al. 2018) train deep networks to predict 2D projections of 3D bounding box vertices, which are then used to infer object poses using a PnP algorithm. These methods often compose of a cascade of multiple stages for object localisation, predicting of box vertices, and pose refinement, are thus time-consuming for inference. What's more, the training and inference of these methods also require the 3D CAD models of the objects. (These work will be further introduced in chapter 6 for the details of how they use the 3D CAD model in their pipeline). In contrast, ours is end-to-end, model-free, and runs in real-time.

3.2.2 Region Proposal Networks

One of main components in the recent successful region-based object detection methods Faster R-CNN (Ren et al. 2015) and in instance segmentation method Mask R-CNN (He et al. 2017) is the Region Proposal Network (RPN). The core

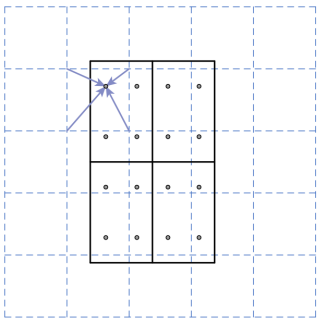


Figure 3.2: RoIAlign Operation. Original figure is from (He et al. 2017). RoIAlign computes the value of pooled feature (solid lines) by bilinear interpolation from the nearby grid points on the feature map (dashed lines).

idea of RPN is to densely and simultaneously score different aspect ratio and scale boxes at each location, keeping the best across the image as proposals (or Region-of-Interest (RoI)).

For each RoI, a fixed-size small feature map (e.g., 7×7) is pooled from the image feature map using the RoIPooling layer (Girshick et al. 2014) or RoIAlign layer (He et al. 2017). These layers work by dividing the RoI into a regular grid and then max-pooling the feature map values in each grid cell. In Faster R-CNN, the outputs of the RoIPooling layer are used to refine the RoI coordinates and to classify the RoI label. The architecture of Faster R-CNN is shown in figure 3.1. In Mask R-CNN, the outputs of the RoIAlign layer(See figure 3.2) are used not only for refining and recognizing the RoI but also for segmenting the object inside the RoI. Although the design of Mask R-CNN is quite simple and straightforward development of Faster R-CNN, it achieves state-of-the-art results on instance segmentation problem. This motivates us to rely on and extend Mask R-CNN for 6dof object pose estimation problem.

3.3 Our Method

The goal of this chapter is to simultaneously detect the 2D positions (represented by bounding boxes), classes, segmentation masks, and the 6dof pose of object instances

in the input image. The first three tasks have been studied well in Mask R-CNN. To achieve a complete system, we add a fourth branch which outputs the 6dof pose. Our model is thus conceptually simple. But the additional 6D pose output is distinct from the other three outputs. It requires a sufficient way to represent the 6dof pose and a careful design of the loss function. In our work, the output of the pose branch is represented by a 4-dimensional vector, in which the first three elements represent the Lie algebra associated with the rotation matrix of the pose; the last element represents the z component of the translation vector of the pose. Given predicted z component and the position of the predicted bounding box, we use projective property to recover the full translation vector.

3.3.1 Mask R-CNN

We start by recapping the Mask R-CNN detector and segmenter (He et al. 2017) in brief. There are two stages in Mask R-CNN. The first is carried out by a Region Proposal Network (RPN), which proposes candidate object bounding boxes (Regions of Interest, RoIs). The second stage then extracts features using RoIAlign from each RoI, and subsequently performs classification, bounding-box regression, and instance segmentation. During training, the multi-task loss on each sampled RoI is

$$L = L_{cls} + L_{box} + L_{mask}. \tag{3.1}$$

Please refer to (He et al. 2017) for loss definitions. RoIAlign layer performs bilinear interpolation over the feature from the RPN, and pools out a fixed-size RoI feature. In analogy to the mask head, our proposed *object coordinate head* learns to transfer from the RoI features to a coordinate map.

3.3.2 Rotation Representation

The choice of representation for the rotation of the pose is critical in our method. The commonly used form are Euler angles, rotation matrix, quaternion *etc.* But they are not suitable for rotation regression because of following reasons.

Euler angles are intuitive due to the explicit meaning of parameters. However, they wrap around at 2π radians. Having multiple values representing the same angle causes difficulty in learning a uni-modal scalar regression task. Furthermore, they suffer from the well-studied issue of gimbal lock (Altmann 2005). Alternatively, a 3×3 orthonormal matrix is usually used to represent the rotation. But it is over-parametrised, and creates the problem of enforcing the orthogonormality constraint when learning through back-propagation. Another common representation is an unit length 4-dimensional quaternion. One of the downsides of quaternion representation is its norm should be unit. This constraint may harm the optimization (Kendall et al. 2015).

In this work, we ultimately choose to use Lie algebra $\mathbf{so}(3)$ to represent the rotation of a 6dof pose. Lie algebra $\mathbf{so}(3)$, whose element is represented as a vector $\boldsymbol{\omega} \in \mathbb{R}^3$, is the tangent space at the identity element of the Lie group $\mathbf{SO}(3)$. Lie group $\mathbf{SO}(3)$ is essentially the space of 3D orthonormal matrix. An element of Lie group $\mathbf{SO}(3)$ represents rotation using a matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$.

Using Lie algebra $\mathbf{so}(3)$ to represent the rotation means that the network only needs to regress three scalars for rotation, without any constraints.

The map from Lie group $\mathbf{SO}(3)$ to Lie algebra $\mathbf{so}(3)$ is defined as *Logarithm map*: $\{\log : \mathbf{SO}(3) \mapsto \mathbf{so}(3)\}$ (Altafini 2001), whose formulation is

$$\boldsymbol{\omega} = [\ln(\mathbf{R})]_\nabla \tag{3.2}$$

$$\ln(\mathbf{R}) = \frac{\theta}{2\sin\theta}(\mathbf{R} - \mathbf{R}^{\mathrm{T}}) \tag{3.3}$$

$$\cos\theta = \frac{tr(\mathbf{R}) - 1}{2} \tag{3.4}$$

where angle $\theta = |\boldsymbol{\omega}|$, $[\ln(\mathbf{R})]_\nabla$ is the 3-vector generated by the skew symmetric matrix $\ln(\mathbf{R})$, using

$$\left[\begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}\right]_\nabla = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.5}$$

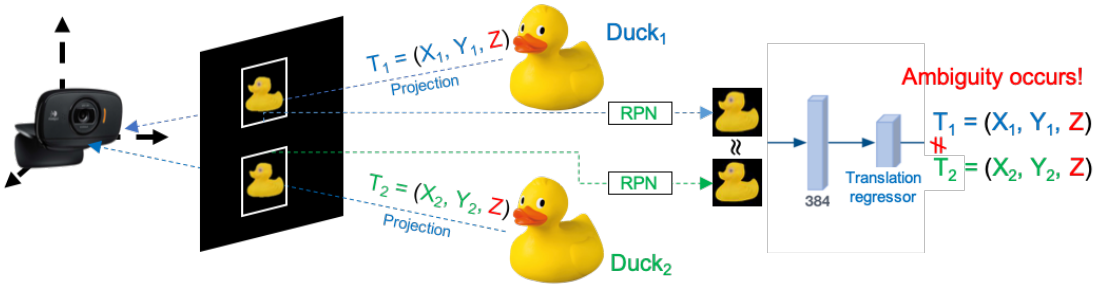


Figure 3.3: Translation ambiguity caused by RPN. Two instances of object duck are projected on image plane from a viewpoint with different translations but share same z-component. Their projected appearances would look alike to a large extent which yields similar RPN features for pose regression. However, the dissimilarity in their translation annotations causes ambiguity to the learning of pose regressor.

During training, we map the groundtruth of rotation matrices in $\mathbf{SO}(3)$ of objects to their associated elements in $\mathbf{so}(3)$ using equation (3.2) to (3.4). These derived groundtruth 3D vectors in $\mathbf{so}(3)$ are the supervision signal for the learning of the 3 rotation parameters in the output of our model.

To recover the rotation matrix from prediction of the network, we use the map from Lie algebra $\mathbf{so}(3)$ to Lie group $\mathbf{SO}(3)$, which is termed as *Exponential map* $\{\exp : \mathbf{so}(3) \mapsto \mathbf{SO}(3)\}$, and is formulated as:

$$\mathbf{R} = e^{\omega} \triangleq e^{[\omega]_\times} = \mathbf{I}_3 + \frac{\sin\theta}{\theta}[\boldsymbol{\omega}]_\times + \frac{1-\cos\theta}{\theta^2}[\boldsymbol{\omega}]_\times^2 \tag{3.6}$$

where $[\boldsymbol{\omega}]_\times$ is a skew symmetric matrix generated by the 3-vector $\boldsymbol{\omega}$ using

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_\times = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}. \tag{3.7}$$

3.3.3 Translation Prediction

Compared to rotation, the representation for translation is straightforward. We learn the translation in the 3D Euclidean space. However, instead of predicting full translation vector with 3 elements, our network is trained to regress only the z component.

The reason is that when (virtually) projecting a 3D model of an object into a 2D image, two translation vectors with the same z and the different x and y

components may produce two objects which have very similar appearance and scale in 2D image. The most notable difference between these two projections is the position of the bounding boxes in the image, which will be however ignored by RPN. This causes difficulty for the pose regressor to predict the x and y components from only appearance information inside the bounding box produced by RPN. See figure 3.3 for illustration. However, the object size and the scale of its textures in the image provide strong cues about the z-coordinate.

In order to recover a full translation vector, we combine the 2D position of bounding box predicted by the bounding box branch in Mask R-CNN and the z component from our pose branch using projective rules. In addition, we assume that the projection of the center of the 3D object model is at the center of the 2D bounding box, since when defining the canonical coordinate systems for interest objects, the origins of them are always placed in the center of their 3D bounding boxes. This is a practically reasonable assumption for the dataset we used in this chapter because no occlusion for objects is considered, neither in the training nor test set, which means the objects are always fully observable from all views. Therefore, the 2D bounding box has a very large IoU with the projections of the 3D bounding box under each view. As a result, the detail formulation for translation estimation is:

$$x = \frac{(u_0 - c_x)z}{f_x} \tag{3.8}$$

$$y = \frac{(v_0 - c_x)z}{f_y} \tag{3.9}$$

where u_0, v_0 are the bounding box center in 2D image, and the f_x, c_x, f_y, c_y are camera intrinsics.

In summary, the pose branch is trained to regress a 4-dimensional vector, in which the first three elements represent rotation part and the last element represents the translation part of the pose.

3.3.4 Multi-task Loss Function

In order to train the network, we define a multi-task loss to jointly train the bounding box class, the bounding box position, the segmentation, and the 6D pose of the object inside the box. Formally, the loss function is defined as:

$$L = L_{cls} + \alpha_1 L_{box} + \alpha_2 L_{mask} + \alpha_3 L_{pose}, \tag{3.10}$$

where L_{cls} is the classification loss, L_{box} is the bounding box regression loss, and L_{mask} is the pixel-wise segmentation loss. L_{pose} is used to train our proposed pose branch. $\alpha_i (i \in 1...3)$ is the loss weight for each loss term respectively.

The pose branch outputs 4 numbers for each RoI, which represents the Lie algebra for the rotation and z component of the translation. To regress the pose, we define pose loss L_{pose} as follows,

$$L_{pose} = \|\boldsymbol{\omega} - \hat{\boldsymbol{\omega}}\|^2 + \beta \|z - \hat{z}\|^2 \tag{3.11}$$

where $\boldsymbol{\omega}$ and $\hat{\boldsymbol{\omega}}$ are two 3-dimensional vectors representing the regressed rotation and groundtruth rotation, respectively; z and $\hat{z}$ are two scalars representing the regressed and groundtruth of translation in z-axis. α is a scale factor to control the rotation and translation regression errors.

3.3.5 Network Architecture

Figure 3.4 shows the schematic overview of our model. Features over the whole image are extracted by the backbone shared between all 4 head branches. For the backbone, we follow Faster R-CNN (Ren et al. 2015) which builds on VGG (Simonyan et al. 2014) with a RPN attached on the last convolutional layer of VGG ($conv5\_3$). A fixed-size 7×7 feature map is pooled for each output RoI of RPN from the $conv5\_3$ feature map using the RoIAlign layer. This pooled feature map is used as input for head branches.

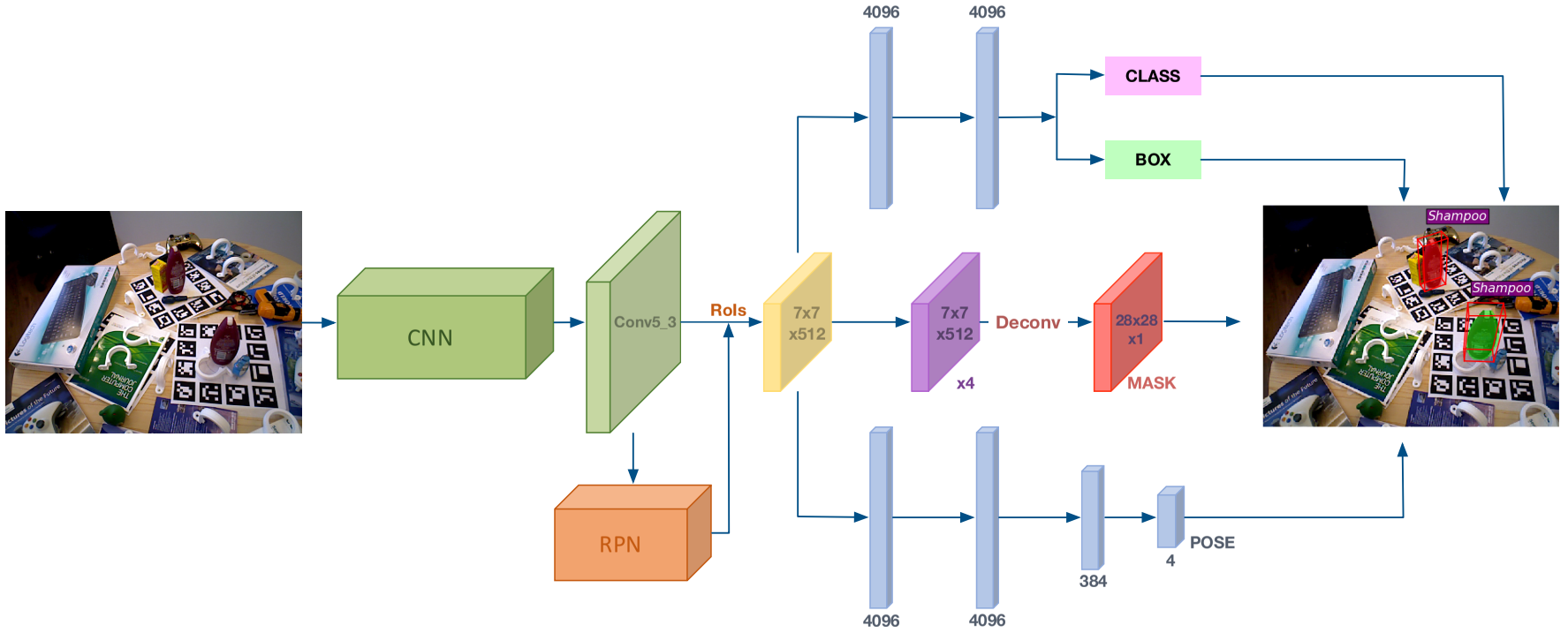


Figure 3.4: An overview of our framework. From left to right: A deep CNN backbone (*i.e.* VGG) is used to extract features over the input image. The RPN is attached on the last convolutional layer of VGG (*i.e.* *conv*5_3) and outputs RoIs. The feature map *conv*5_3 are extracted and pooled into a fixed size 7×7 for each RoI, and used as inputs for 4 head branches. There are 4 fully connected layers in the pose head. The last fully connected layer outputs 4 numbers which represent for the pose. As shown on the right image, the network outputs the detected object instances, the predicted classes (i.e., Shampoo), the predicted segmentation masks (different object instances are shown with different colours) and the predicted 6D poses (shown with 3D boxes).

The head branches aim to solve 4 different tasks, *i.e.*, bounding box prediction, bounding box classification, instance segmentation, and 6dof pose estimation for the object inside the box. The structures of first three heads are inherited from Mask R-CNN (He et al. 2017). A small modification is made to segmentation head to adapt our application. We use four convolutional layers with kernel size 3 with Relu as the feature extractor. The feature map is then upsampled to 28×28 by a deconvolutional layer, whose output is the segmentation prediction.

In order to use shared features for our goal of pose estimation, we firstly flatten the RoI feature into a vector and then regress the 4D vector from it. The proposed pose head branch consists of a sequence of 4 fully connected layers followed by ReLU (except for the last one). Their number of filters are $4096 \rightarrow 4096 \rightarrow 384 \rightarrow 4$. This design is simple but sufficient enough to achieve good accuracy for pose prediction.

3.3.6 Training and Inference

Training: Our model is implemented using Caffe library (Jia et al. 2014). We resize all the input RGB image to a fixed size 480×640. We use 5 scales and 3 aspect ratios for the size of RoIs, resulting 15 anchors in the RPN. The 5 scales are 16×16, 32×32, 64×64, 128×128 and 256×256 and the 3 aspect ratios are 2, 1 and 0,5). This design allows the network to detect small objects.

We train the network in an end-to-end manner using stochastic gradient descent with 0.9 momentum and 0.0005 weight decay. The network is trained for $250k$ iterations with batch size 1. The learning rate is set to 0.001 for the first $150k$ iterations and then decreased by 10 for the remaining iterations. The top 2000 RoIs from RPN (with a ratio of 1:3 of positive to negative) are subsequently used for computing the multi-task loss. A RoI is considered positive if it has an intersection over union (IoU) with a groundtruth box of at least 0.5 and negative otherwise. The losses L_{mask} and L_{pose} are defined for only positive RoIs.

For the hyperparamaters in our loss term, we set α_1 to α_4 in (3.10) are to 1, 1, 2, 2, respectively. β in (3.11) is empirically set to 1.5.

Inference: At the inference time, the top 1000 RoIs produced by the RPN are selected and fed into the object detection and classification branches, followed by non-maximum suppression (Girshick 2015). From the outputs of the detection branch, we select the output boxes that have classification scores higher than a certain threshold (in our case is 0.9) as the detection results. The segmentation and the pose branches are then applied on the detected boxes, which output segmentation masks and the 6dof poses for the objects inside the boxes.

3.4 Experiments

3.4.1 Datasets

We evaluate our method on two datasets. First, we evaluate our system on the single object pose estimation benchmark: LINEMOD (Hinterstoisser et al. 2012) dataset. The images in each of 13 object sequences contain multiple objects, however, only one interest object is annotated with the groundtruth class label, bounding box, and 6D pose. The camera intrinsic matrix is also provided with the dataset. Using the given groundtruth 6D poses, the object CAD models, and the camera matrix, we can also compute the groundtruth segmentation mask for the annotated objects.

One problem arises in the original LINEMOD dataset since our goal is to train a unified pose regression model for all objects. Interest object A in sequence 1 is considered as background in sequence 2, whose interest object is B. The inconsistent labelling for object A in different sequences cause confusion to the network when the input data includes images from sequence 1 and 2 (and also in 3-13) and therefore hinders the training of object detection. To deal with that, for each object sequence, we use RefineNet by Lin et al. 2017, a state-of-the-art semantic segmentation algorithm, to train a semantic segmentation model for each object.

The trained model for object A using sequence 1 is applied on all the rest sequences. The predicted masks for object A in other sequences are then filtered out from the background, so that the presences of objects without annotated information does not affect the training. 30% of the images from each sequence for are selected for training and validation. We perform evaluation on the remaining images.

Then our method is then performed on the dataset with multiple object instances provided by Tejani et al. 2014. It consists of six object sequences. Each sequence contains images that have multiple instances of the same object in different viewpoints. The object is provided with the groundtruth class label, bounding box, and 6D pose. Using the given groundtruth 6D poses, the object models, and the known camera matrix, we are able to compute the groundtruth segmentation masks for object instances. Although provided with depth, we only use RGB images in our experiment. We randomly split 50% images in each sequence for training and evaluation. The remaining images serve as the test set.

3.4.2 Evaluation Metrics

The metrics we use to assess the object pose estimation performance are 2D-projection, ADD-10 and 5cm5deg. 2D-projection metric measures pose errors in 2D, in which we project the 3D object model into the image using the ground truth pose and the estimated pose. The estimated pose is accepted if the average reprojection error of all points is below 5px. ADD is the average 3D distance of model points transformed by the predicted pose and ground truth pose. For symmetric objects, ADD is relaxed to ADD-S, which is the distance between the closest points in two transformed models. If the average (or closest) distance derived by a test pose is less than 10% of the object diameter, the pose estimate is considered correct. As the for 5cm5deg, an estimate is correct when the translation and rotation error is below $(5cm, 5°)$.

We also evaluate the detection and segmentation results. A detection / segmentation is true-positive if its IoU with the groundtruth box / segmentation mask is higher than a threshold.

3.4.3 Single Object Pose Estimation

We first evaluate our method on 2D recognition task, including detection and segmentation. Table 3.1 presents the 2D detection and segmentation results at different IoU. At an IoU 0.5, both detection and segmentation achieve nearly perfect scores for all object categories. Even at a more challenging IoU 0.9, although the accuracies decrease, the detection and segmentation branches still perform quite well with average detection score 90%.

Secondly we evaluate the performance 6dof pose estimation, which is the main focus of this chapter. We compare our method against the state-of-the-art RGB based 6D object pose estimation methods such as BB8 (Rad et al. 2017), SSD-6D (Kehl et al. 2017), Brachmann et al. 2016, and Tekin (Tekin et al. 2018).

Table 3.2 reports the comparative pose estimation accuracies between ours and the state-of-the-art work (Brachmann et al. 2016; Rad et al. 2017; Kehl et al. 2017; Tekin et al. 2018). All the methods only use RGB images as inputs to predict the poses. Note that except ours and Tekin et al. 2018, other methods comprise of multiple-stages including a 2D object detection, an initial pose estimation, and a pose refinement.

One can see that our method significantly outperforms all the considered competitors when they are used without a post-refinement under all evaluation metrics. The improvements are more significant when the errors are computed using the estimated poses directly (i.e., ADD and 5cm 5°), and less significant when evaluated in 2D using IoU. Even when the competitor methods such as (Rad et al. 2017) and (Kehl et al. 2017) further refine their estimated poses using a post-refinement step, our method is still very competitive. Note that the post-refinement

	Ape	Bvise	Cam	Can	Cat	Driller	Duck	Box	Glue	Holep	Iron	Lamp	Phone	Average
	IoU 0.5													
2D-Detection	99.8	100	99.7	100	99.5	100	99.8	99.5	99.2	99.0	100	99.8	100	99.7
2D-Segmentation	99.5	99.8	99.7	100	99.1	100	99.4	99.5	99.0	98.6	99.2	99.4	99.7	99.4
	IoU 0.9													
2D-Detection	85.4	91.7	93.3	93.6	89.3	87.5	86.3	94.2	81.1	93.2	92.5	91.3	90.8	99.3
2D-Segmentation	80.6	57.0	91.4	62.5	52.1	74.6	81.2	91.9	73.3	84.6	90.3	85.0	84.6	77.6

Table 3.1: 2D detection and segmentation results on LINEMOD dataset for single object.

	Ape	Bvise	Cam	Can	Cat	Driller	Duck	Box	Glue	Holep	Iron	Lamp	Phone	Avg.
	2D-projection metric													
Ours	99.8	100	99.7	100	99.2	100	99.8	99.0	97.1	98.0	99.7	99.8	99.1	99.3
Tekin et al. 2018	99.8	99.9	100	99.8	99.9	100	100	99.9	99.8	99.9	100	100	100	99.9
Brachmann et al. 2016(*)	98.2	97.9	96.9	97.9	98.0	98.6	97.4	98.4	96.6	95.2	99.2	97.1	96.0	97.5
Kehl et al. 2017(*)	99.0	100	99.0	100	99.0	99.0	98.0	99.0	98.0	99.0	99.0	99.0	100	99.1
	5cm5° metric													
Ours	57.8	72.9	75.6	70.1	70.3	72.9	67.1	68.4	64.6	70.4	60.7	70.9	69.7	68.5
Brachmann et al. 2016(*)	34.4	40.6	30.5	48.4	34.6	54.5	22.0	57.1	23.6	47.3	58.7	49.3	26.8	40.6
Rad et al. 2017(*)	80.2	81.5	60.0	76.8	79.9	69.6	53.2	81.3	54.0	73.1	61.1	67.5	58.6	69.0
	ADD metric													
Ours	38.8	71.2	52.2	86.1	66.2	82.3	32.5	79.4	63.7	56.4	65.1	89.4	65.0	65.2
Rad et al. 2017	27.9	62.0	40.1	48.1	45.2	58.6	32.8	40.0	27.0	42.4	67.0	39.9	35.2	43.6
Kehl et al. 2017	0	0.18	0.41	1.35	0.51	2.58	0	8.9	0	0.30	8.86	8.20	0.18	2.42
Tekin et al. 2018	21.6	81.8	36.6	68.6	41.8	63.5	27.2	69.6	80.0	42.6	75.0	71.1	47.7	55.9
Brachmann et al. 2016(*)	33.2	64.8	38.4	62.9	42.7	61.9	30.2	49.9	31.2	52.8	80.0	67.0	38.1	50.2
Rad et al. 2017(*)	40.0	91.8	55.7	64.1	62.6	74.4	44.3	57.8	41.2	67.2	84.7	76.5	54.0	62.7
Kehl et al. 2017(*)	-	-	-	-	-	-	-	-	-	-	-	-	-	76.3

Table 3.2: Pose estimation accuracy on LINEMOD dataset for single object. (*) indicates methods used with post-refinements

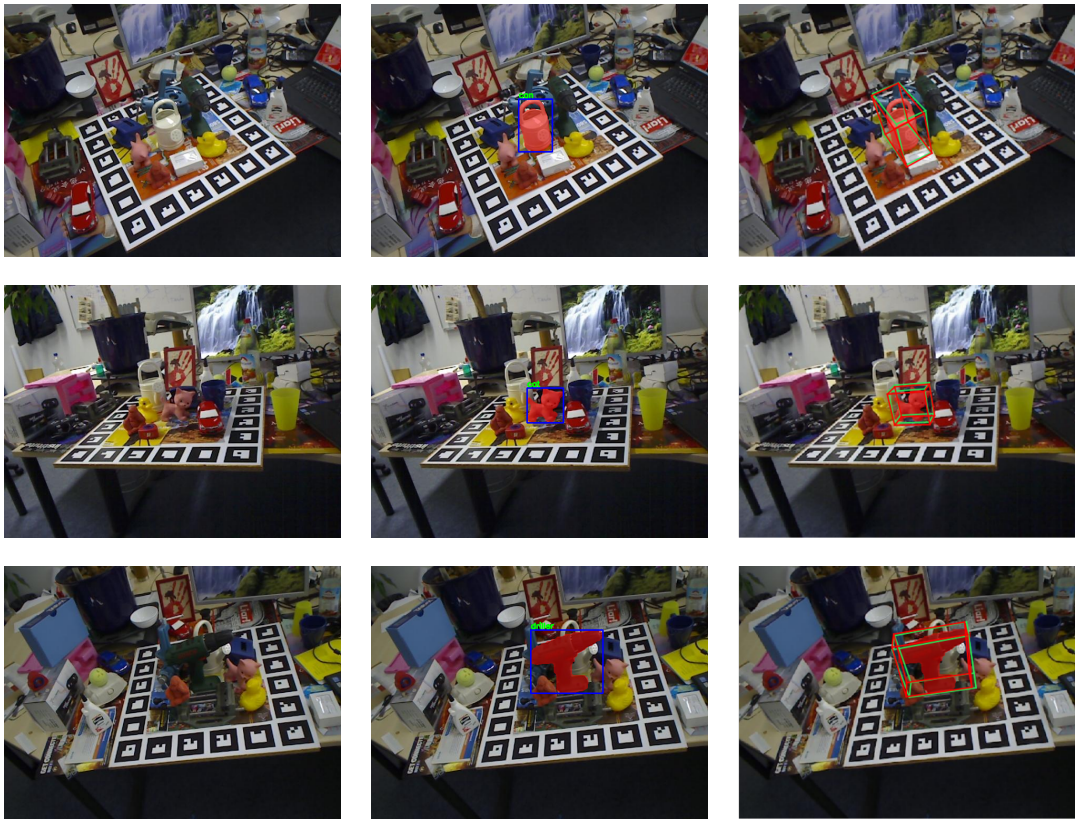


Figure 3.5: Qualitative results for single object pose on LINEMOD dataset. From left to right: (i) original images, (ii) the predicted 2D detections, classes, and segmentation (different instances are shown with different colours), (iii) 6D poses in which the green boxes are groundtruth poses and the red boxes are predicted poses. Best view in colour.

cost is often expensive, for instance the method in (Brachmann et al. 2016) takes about 100ms per object. Figure 3.5 shows some qualitative results of our method for single object pose estimation on LINEMOD dataset.

3.4.4 Multiple Object Instance Pose Estimation

The 2D detection and segmentation results on the dataset of Tejani et al. 2014 are shown in table 3.3. At an IoU 0.5, we achieve nearly perfect scores. Then at IoU 0.9, the accuracy decreases slightly. We found that the Shampoo category gives the most drop. It is caused by its flat shape, e.g., at some certain poses, the projected 2D images only contain a small side edge of the object, resulting the drop of scores at high IoU.

	Camera	Coffee	Joystick	Juice	Milk	Shampoo	Avg.
				IoU 0.5			
2D-Detection	99.8	100	99.8	99.2	99.7	99.5	99.6
2D-Segmentation	99.8	99.7	99.6	99.0	99.3	99.5	99.4
				IoU 0.9			
2D-Detection	93.3	98.2	98.7	94.8	94.5	87.3	94.4
2D-Segmentation	88.3	97.0	98.1	91.2	91.6	82.0	91.3

Table 3.3: 2D detection and segmentation results on the dataset of Tejani et al. 2014 for multiple object instances.

	Camera	Coffee	Joystick	Juice	Milk	Shampoo	Avg.
				2D-projection			
Ours	99.2	100	99.6	98.4	99.5	99.1	99.3
Kehl et al. 2017	97.3	99.8	100	99.4	97.0	99.3	98.8
				5cm5° metric			
Ours	76.5	18.7	60.2	85.6	73.5	72.4	64.5
				ADD metric			
Ours	80.4	35.4	27.5	81.2	71.6	75.8	62.0

Table 3.4: Pose estimation accuracy on the dataset of Tejani et al. 2014 for multiple object instances.

The pose estimation accuracies are reported in table 3.4. We note that except SSD-6D (Kehl et al. 2017), none of the previous RGB based pose object estimation works report their results using this data. Also SSD-6D only reports their pose accuracies using the 2D-projection metric. It can be seen from the table 3.4 that our method performs better than SSD-6D even it uses a post-refinement. Furthermore, under more tricky 5cm 5° and ADD metrics, our method still achieves impressive results with mean accuracies 64.5% and 62.0%, respectively. Figure 3.6 shows the qualitative results for the predicted bounding boxes, classes, segmentation, and 6D poses for multiple object instances.

The results in table 3.4 show that the Coffee sequence has the lowest score. We found that it is because the nearly rotational symmetry (in both shape and texture) of that sequence. By the symmetric rotation, any rotation of 3D object in the Yaw angle will produce the same object appearance in the 2D image. This causes the

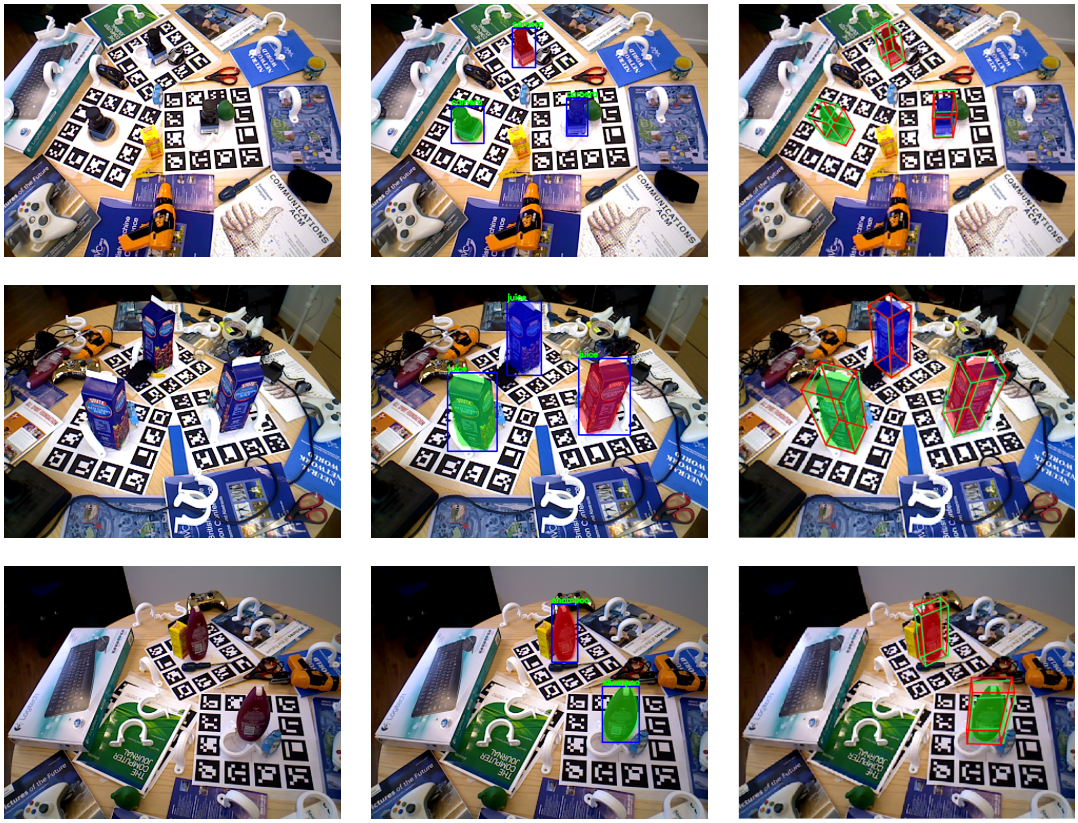


Figure 3.6: Qualitative results for pose estimation on the multiple object instance dataset of Tejani et al. 2014. From left to right: (i) the original images, (ii) the predicted 2D detections, classes, and segmentation (different instances are shown with different colours), (iii) 6D poses in which the green boxes are groundtruth poses and the red boxes are predicted poses. Best view in colour.

network to be confused when predicting rotation using only appearance information.

3.4.5 Timing

To perform the single object pose estimation, the end-to-end architecture of our model allows the inference to run at 10fps on a Titan X GPU. It is worth noting that, although our design is not optimized for speed, it runs several times faster than BB8 (Rad et al. 2017) (3fps) and Brachmann et al. 2016 (2fps), and comparable with SSD-6D (Kehl et al. 2017). However, these methods report their running times using the LINEMOD dataset, which contains only one object instance in each image. Due to the post-refinement, their computational cost will increase rapidly when tested on images with multiple object instances such as the dataset

of Tejani et al. 2014. In contrast, the running time of our method stays almost the same regardless of the number of object instances.

3.5 Conclusion

In this chapter we have presented an end-to-end architecture for estimating 6dof object poses from a single RGB image. By extending the recent Mask R-CNN architecture with a new pose head, we train a multi-task network which can simultaneously recognize, segment, and recover 6dof pose of the object. The novel pose head branch uses Lie algebra based rotation representation, which is well suited for deep regression. We recover the translation from the z component given by pose head and the position of bounding box from detection head. Our method outperforms the most of the RGB-based 6dof object pose estimation methods when they are all used without post-refinements. Furthermore, our method also allows a fast inference which is around 10 fps, which meets the requirement for real time application.

Bibliography

Gordon, Iryna and David G Lowe (2006). "What and where: 3D object recognition with accurate pose". In: *Toward category-level object recognition*. Springer, pp. 67–82.

Martinez, Manuel, Alvaro Collet, and Siddhartha S Srinivasa (2010). "Moped: A scalable and low latency object recognition and pose estimation system". In: *2010 IEEE International Conference on Robotics and Automation*. IEEE, pp. 2043–2049.

Wagner, Daniel, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg (2008). "Pose tracking from natural features on mobile phones". In: *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE, pp. 125–134.

Lowe, David G (2004). "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2, pp. 91–110.

Hinterstoisser, Stefan, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab (2012). "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes". In: *Asian conference on computer vision*. Springer, pp. 548–562.

Rios-Cabrera, Reyes and Tinne Tuytelaars (2013). "Discriminatively trained templates for 3d object detection: A real time scalable approach". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2048–2055.

Brachmann, Eric, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother (2014). "Learning 6d object pose estimation using 3d object coordinates". In: *European conference on computer vision*. Springer, pp. 536–551.

Krull, Alexander, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother (2015). "Learning analysis-by-synthesis for 6D pose estimation in RGB-D images". In: *Proceedings of the IEEE international conference on computer vision*, pp. 954–962.

Michel, Frank, Alexander Kirillov, Eric Brachmann, Alexander Krull, Stefan Gumhold, Bogdan Savchynskyy, and Carsten Rother (2017). "Global hypothesis generation for 6D object pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 462–471.

Girshick, Ross (2015). "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*, pp. 91–99.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Kendall, Alex, Matthew Grimes, and Roberto Cipolla (2015). "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946.

Kendall, Alex and Roberto Cipolla (2017). "Geometric loss functions for camera pose regression with deep learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5974–5983.

Prisacariu, Victor A and Ian D Reid (2012). "PWP3D: Real-time segmentation and tracking of 3D objects". In: *International journal of computer vision* 98.3, pp. 335–354.

Tejani, Alykhan, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim (2014). "Latent-class hough forests for 3D object detection and pose estimation". In: *European Conference on Computer Vision*. Springer, pp. 462–477.

Brachmann, Eric, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother (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*, pp. 3364–3372.

Rad, Mahdi and Vincent Lepetit (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*, pp. 3828–3836.

Kehl, Wadim, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab (2017). "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1521–1529.

Tekin, Bugra, Sudipta N Sinha, and Pascal Fua (2018). "Real-time seamless single shot 6d object pose prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 292–301.

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Altmann, Simon L (2005). *Rotations, quaternions, and double groups*. Courier Corporation.

Altafini, Claudio (2001). "The de Casteljau algorithm on SE (3)". In: *Nonlinear control in the year 2000*. Springer, pp. 23–34.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Jia, Yangqing, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell (2014). "Caffe: Convolutional architecture for fast feature embedding". In: *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678.

Lin, Guosheng, Anton Milan, Chunhua Shen, and Ian Reid (2017). "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1925–1934.

4

GPoseNet: A Hybrid Probabilistic Model for Camera Relocalisation

Contents

We introduce the method that estimates the predictive distribution for pose estimated from a deep model in this chapter. The main idea is to use the embedded

features of image from CNNs to enable probabilistic Gaussian Process Regression (GPR) for uncertainty estimation. We first provide a brief explanation to GPR, as well as Stochastic Variational Inference (SVI) for model simplification. The method that combines CNNs and SVI GPR is detailed subsequently, which leverages the deterministic features learnt by CNNs to formulate the GPR. In addition, we use a kernel function for vector valued function to deal with the inner correlation between the elements of translation and rotation, respectively. This work was presented at the British Machine Vision Conference 2018.

4.1 Introduction

In the previous chapter, we proposed a method that directly regresses the 6dof pose for all interest objects in an image. Fundamentally, it acts like the absolute camera pose regression method PoseNet (Kendall et al. 2015), but instead of using the whole image as input for pose regressor, our model extracts useful features only inside object bounding boxes to decide pose. Despite this key difference, the essence of both methods is to use the power of CNNs and all the training data to build a map from image to pose space directly. In section 2.3.1 we defined this kind of method as 'regression-based'.

These regression-based approaches are the proof of concept of using deep neural networks for the task of pose estimation. Although the results are appealing, a significant factor has however missed that it is not straightforward – or even well understood – how to model the uncertainty of CNN outputs.

Exploiting uncertainty in deep neural networks is an eye-catching topic in learning community. The probability distribution of the prediction from a deep perception system can be used in a variety of ways. Most notably for our purposes, the distribution over a pose regression result can be interpreted as its uncertainty, making the result amenable for use within the standard data fusion algorithms such

as a Kalman Filter for state estimation in SLAM systems (Davison et al. 2007) to improve the accuracy of localization over time.

In this chapter, we aim to model the uncertainty of the regression model that directly predicts the camera pose from a single RGB image. Closely related work has been done in Bayesian PoseNet (Kendall et al. 2016), the probabilistic version of PoseNet (Kendall et al. 2015). The architecture and training process of Bayesian PoseNet are exactly same with PoseNet. But during inference, the authors keep the existence of *dropout* layers – i.e, different stochastic connections between neurons – in the trained model. Thanks to these dropout layers before the regressors, multiple (and different, because of randomness of the dropout) samples of camera pose are obtained during inference with repetitive forwards using a same test image. Then the distribution of the 6dof pose can be empirically summarized by these pose samples. The mathematics behind this approximation has been well studied in (Gal et al. 2016). However, this distribution-from-samples method is not very resource-friendly, requiring a separate forward inference per sample.

Compare to CNNs, Gaussian Process Regression (GPR) (Williams et al. 2006) is a probabilistic model that inherently provides a tractable predictive distribution for the output. But when one applies GPR to real world applications that involve large scale datasets, the low efficiency prevents GPR from being plug-and-play. This arises from four factors. First, GPR scales as a power of the size of the training set n. The matrix inverse for computing precision matrices has complexity $O(n^3)$, which is prohibitive because n is usually a large number in vision tasks. Second, even with complexity reduction, the training of sparse GPR still involves all training samples in each optimization step. Third, the high-dimensional image data is not a straightforward input for GPR's kernel function. Forth, GPR is less commonly used for vector valued functions. In the learning-based pose regression, rotation and translation are both represented as vectors which are non-trivial for the design of kernel function.

A solution to the first two factors is provided by Stochastic Variational Inference (SVI) (Hensman et al. 2013), which treats the mean and covariance of a lower dimensional variational posterior as the global parameters, turning the variational GPs into a "parametric" model. To solve the third issue mentioned above, we use the feature vectors learnt from a CNN to describe the input image. For vector valued outputs like translation and rotation, the *matrix valued* kernel matrix (*i.e.*, matrix of matrices) is used to maintain the correlation between the elements of a vector valued function. Specifically, different from the scalar valued kernel matrix, whose elements are scalars, an element of the kernel matrix for vector-valued function is a $D \times D$ positive semi-definite matrix, where D is the length of the output vector. This matrix describes the covariance between the elements of the vector-valued function.

With these tools ready to use, our main contribution is to show how to combine CNNs and GPR naturally, proposing a probabilistic framework to model the uncertainty in the regression of 6DoF camera pose based on a RGB image, while overcoming the complexity issues of naive GPR. We exploit the CNN to extract discriminative features and use the GPR to perform probabilistic inference. We show that the mean of our predictive pose distribution has the comparable accuracy to the state-of-the-art pure RGB based method for camera localization, and meanwhile the covariance is compatible with the uncertainty from Bayesian PoseNet, but with significant computational resource saving.

To achieve this combination we build an objective function for the whole framework that aims to minimize two Kullback-Leibler (KL) divergences between distributions. In the original PoseNet, the importance between rotational and translational penalization are balanced by grid-search over the network hyperparameters. Later, to avoid hyperparameters tuning, Kendall et al. 2017 embedded the translation and rotation into a single photometric loss via geometrical transformation and affine projection. However this is not a universal solution for some other multi-task CNNs that lack of underlying connections between tasks. Our use of the

KL-divergences not only results in a loss function that permits network optimisation in an end-to-end fashion, but furthermore, as shown in section 4.3.5, it leads to greatly improved robustness to the choice of hyperparameters, obviating the need for expensive grid search during training.

To the best of our knowledge, this is the first work that combines the CNN and GPR to perform probabilistic inference for large scale computer vision task.

4.2 Background

4.2.1 Uncertainty in Camera Relocalisation

Since publications for camera relocalisation have been reviewed thoroughly in chapter 2, we address the methods that investigate uncertainty learning in publications that involve CNNs and GPRs.

In the paper of Gal et al. 2016, the authors prove that the widely-used dropout technique can be mathematically viewed as an approximation to the posterior of the deep Gaussian Process (Damianou et al. 2013). By running same test point through the model multiple times with the existence of dropout (which is often deactivated in most deterministic CNN models at inference time), the different connections between neurons lead to a set of Monte Carlo samples from the approximated variational posterior over the output. The Bayesian PoseNet (Kendall et al. 2016) is a well-demonstrated application of dropout bayesian approximation. Without changing the training pipeline of PoseNet, it brings the camera relocalisation to a probabilistic level. Whilst it improves the accuracy of the pose estimation, the uncertainty can be also obtained empirically with grounded theoretical support from publication by Gal et al. 2016.

Beyond the success of two versions of PoseNet, two problems can be further discussed. The first one is the computational efficiency of Bayesian PoseNet. To generate accurate posteriors for camera pose, one often needs many prediction

samples, which results in great computational cost. The second concerns the network hyperparameters. Due to the different units and scales of the two distinct quantities of pose – translation and rotation, the choice of the hyperparameters that balance the loss terms in the final learning objective is therefore challenging. Experimentally we also found that they are heavily scene-dependent, and the optimal setting takes much effort to find. To this end, we propose to use the combination of CNN and GPR to improve the efficiency and eliminate hyperparameter tuning. A comprehensive explanation is given in section 4.3.5.

4.2.2 Gaussian Process Regression

Gaussian process regression is a fully Bayesian non-parametric model that elegantly estimates the posterior distribution of the target function. The introduction of GPR for 1-d functions is given in this section, bringing in the notation used in this chapter. Let $\{\mathbf{x}_i, y_i\}_{i=1}^n$ denote the whole dataset. $\mathbf{x}_i \in \mathbf{X}$ is a sample from all points $\mathbf{X}$ in the feature domain $\mathbb{R}^F$, and $y_i \in \mathbf{y}$ is the observation of a function f at point $\mathbf{x}_i$ with independent Gaussian noise σ^2.

A Gaussian Process is a Gaussian distribution over functions (Williams et al. 2006). Mathematically, a Gaussian process is specified by a mean function and covariance function

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{4.1}$$

where the mean and covariance are

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{4.2}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))^T] \tag{4.3}$$

To simplify the notation, the mean function is usually set to zero. The covariance of the prior is based on a kernel function k, which essentially describes the

similarity between input points. The most commonly used kernel function is squared exponential (or Radial Basis Function, RBF), which has the form of

$$k(x, x') = \exp(-\frac{1}{2l^2}(x - x')^2), \tag{4.4}$$

where l is length-scale.

In order to take the possible noise for training data about the function into consideration, an independent Gaussian noise σ^2 is added to the distribution

$$\mathbf{y} \sim \ \mathcal{N}(\mathbf{0}, K(\mathbf{X}, \mathbf{X}') + \sigma^2 \mathbf{I}_n) \tag{4.5}$$

Given a test point $\mathbf{x}^\star$, the inference also take place directly in the space of functions. The function of $\mathbf{x}^\star$, $f\star$ assumed to follow the same distribution as the training data, meaning that the joint distribution of test data and training data is given by:

$$\begin{bmatrix} \mathbf{y} \\ f^\star \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}') + \sigma^2 \mathbf{I}_n & K(\mathbf{X}, \mathbf{x}^\star) \\ K(\mathbf{x}^\star, \mathbf{X}) & K(\mathbf{x}^\star, \mathbf{x}^\star) \end{bmatrix}\right) \tag{4.6}$$

The conditional posterior of the test functions $f^\star$ can be inferred via multivariate Gaussian theorem (Williams et al. 2006):

$$p\left(f^\star|\mathbf{y}\right) = \mathcal{N}\left(f^\star|\mathbf{K}_{\star n}(\mathbf{K}_{nn} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \mathbf{K}_{\star\star} - \mathbf{K}_{\star n}(\mathbf{K}_{nn} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{n\star}\right). \tag{4.7}$$

where $\mathbf{K}_{\star n} = K(\mathbf{x}^\star, \mathbf{X})$, $\mathbf{K}_{nn} = K(\mathbf{X}, \mathbf{X}')$, $\mathbf{K}_{n\star} = K(\mathbf{X}, \mathbf{x}^\star)$ and $\mathbf{K}_{\star\star} = K(\mathbf{x}^\star, \mathbf{x}^\star)$

However, the exact posterior requires $O(n^3)$ to compute, where n is the size of training set. It hinders the efficiency of GPR.

4.2.3 Sparse Gaussian Process Regression Approximation

A group of sparse GPs that aim to reduce the complexity of the full GPs are well-researched, such as Deterministic Training Conditional Approximation(DTC) (Csató et al. 2002), Fully Independent Training Conditional Approximation (FITC) (Snelson et al. 2006) and Partially Independent Training Conditional Approximation (PTIC) (Quiñonero-Candela et al. 2005). These methods approximate the prior

using a set of pseudo data points and perform the exact inference. They find the optimal sparse point set by maximizing the likelihood according to the observations. The computational complexity downscales to the to $O(nm^2)$, where m is the number of the pseudo training points.

One disadvantage of prior approximation methods is that when new training data comes, action has to be taken to renew the number or positions of the pseudo points. In other words, they are unnatural from a generative modelling perspective.

4.2.4 Variational Free-Energy (VFE) Method

In contrast, VEF (Titsias 2009) achieves model simplification by approximating the exact posterior $p(\mathbf{u}|\mathbf{y})$ with a variational distribution $q(\mathbf{u})$, where $\mathbf{u}$ is defined as the support or inducing variables, which represents the pseudo training points.

Denote by $\mathbf{Z} \in \mathbb{R}^{m \times F}$ the inducing points with number of m, where F is the dimension of the inputs. The optimal inducing points $\hat{\mathbf{Z}}$ can be found via maximizing the variational lower bound (ELBO) of the log of marginal likelihood $p(\mathbf{y})$.

The log of marginal likelihood and its exact lower bound is

$$\log p(\mathbf{y}) = \log \int p(\mathbf{y}|\mathbf{u}, \mathbf{f})\, p(\mathbf{u}, \mathbf{f})\, d\mathbf{f} d\mathbf{u} \tag{4.8}$$

$$\geq \int p(\mathbf{u}|\mathbf{f}) \log p(\mathbf{y}|\mathbf{f}) \frac{p(\mathbf{f}|\mathbf{u})\, p(\mathbf{u})}{p(\mathbf{u}|\mathbf{f})} d\mathbf{f} d\mathbf{u}. \tag{4.9}$$

Since the exact posterior of inducing variables $p(\mathbf{u}|\mathbf{f})$ in 4.9 still needs to compute $(\mathbf{K}_{nn} + \sigma^2 \mathbf{I})^{-1}$, Titsias 2009 approximates it with a variational distribution q over inducing variables $\mathbf{u}$, and the variational lower bound becomes

$$\log p(\mathbf{y}) \geq \mathcal{L}(q, \mathbf{Z}) \tag{4.10}$$

$$= \int q(\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) \frac{p(\mathbf{f}|\mathbf{u})\, p(\mathbf{u})}{q(\mathbf{u})} d\mathbf{f} d\mathbf{u} \tag{4.11}$$

$$= \int q(\mathbf{u}) \left\{ \mathbb{E}_{<p(\mathbf{f}|\mathbf{u})>} (\log p(\mathbf{y}|\mathbf{f})) + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right\} d\mathbf{u} \tag{4.12}$$

where

$$\mathbb{E}_{<p(\mathbf{f}|\mathbf{u})>}\left(\log p\left(\mathbf{y}|\mathbf{f}\right)\right) = \log\mathcal{N}(\mathbf{y}|\mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{u}, \sigma^2\mathbf{I}) - \frac{1}{2}\sigma^{-2}Tr(\mathbf{K}_{nn} - \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}) \tag{4.13}$$

is the lower bound of log of conditional likelihood $p\left(\mathbf{y}|\mathbf{u}\right)$.

After integrating variable $\mathbf{u}$, Titsias 2009 proves the conclusion that the final formulation of the ELBO to $\log p\left(\mathbf{y}\right)$ is

$$\mathcal{L}(\mathbf{Z}) = \log\mathcal{N}\left(\mathbf{y}|\mathbf{0}, \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn} + \sigma^2\mathbf{I}\right) - \frac{1}{2\sigma^2}Tr\left(\mathbf{K}_{nn} - \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}\right), \tag{4.14}$$

and the optimal variational posterior $q^*(\mathbf{u})$ is with mean and covariance:

$$\boldsymbol{\mu} = \sigma^{-2}\mathbf{K}_{mm}\Sigma^{-1}\mathbf{K}_{mn}\mathbf{y} \tag{4.15}$$

$$\Lambda = \mathbf{K}_{mm}\Sigma^{-1}\mathbf{K}_{mm}, \tag{4.16}$$

where $\Sigma = \mathbf{K}_{mm} + \sigma^{-2}\mathbf{K}_{mn}\mathbf{K}_{nm}$. The complexity is now $O(nm^2)$.

4.2.5 Stochastic Variational Inference (SVI)

Note that when computing $\mathcal{L}(\mathbf{Z})$ and $q^*(\mathbf{u})$ during optimization, the existence of $\mathbf{K}_{nm}$ (or $\mathbf{K}_{mn}$) and $\mathbf{y}$ makes the algorithm need to use all training samples. This is disadvantageous for tasks with large datasets.

In SVI for GPs, Hensman et al. 2013 propose to use a parametric variational Gaussian posterior for $\mathbf{u}$, such that the mean and covariance of $q_g(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m}, \mathbf{S})$ act as the *global* variational parameters across all training samples. This enables the joint training of $\mathbf{m}$, $\mathbf{S}$ and $\mathbf{Z}$ via batch data to perform SGD. The lower bound of the log marginal likelihood then changes from equation (4.12) to

$$\begin{aligned}
\mathcal{L}_{svi}(\mathbf{m}, \mathbf{S}, \mathbf{Z}) &= \int q_g\left(\mathbf{u}\right)\left\{\mathbb{E}_{<p(\mathbf{f}'|\mathbf{u})>}\left(\log p\left(\mathbf{y}'|\mathbf{f}'\right)\right) + \log\frac{p(\mathbf{u})}{q_g\left(\mathbf{u}\right)}\right\}d\mathbf{u} \\
&= \mathbb{E}_{<q_g(\mathbf{u})>}\left(\mathbb{E}_{<p(\mathbf{f}'|\mathbf{u})>}\left(\log p\left(\mathbf{y}'|\mathbf{f}'\right)\right) + \log p(\mathbf{u}) - \log q_g\left(\mathbf{u}\right)\right) \\
&= \log\mathcal{N}(\mathbf{y}'|\mathbf{K}_{n'm}\mathbf{K}_{mm}^{-1}\mathbf{m}, \sigma^2\mathbf{I}) - \frac{1}{2\sigma^2}Tr\left(\mathbf{K}_{n'n'} - \mathbf{K}_{n'm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn'}\right) \\
&\quad - \frac{1}{2}Tr\left(\mathbf{S}\boldsymbol{\Lambda}'^{-1}\right) - \mathrm{KL}(q_g(\mathbf{u})||p(\mathbf{u}))
\end{aligned} \tag{4.17}$$

where $\mathrm{KL}(q_g(\mathbf{u})||p(\mathbf{u}))$ is the KL divergence between the variational posterior and the exact prior $p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K}_{mm})$. Note that in equation (4.17), $(\cdot)'$ means that it is from/for the batch data. We use this lower bound as the objective of our GP regressors.

4.3 Modelling Uncertainty for Camera Relocalisation

4.3.1 Problem Formulation

Given a RGB image $I_i \in \mathcal{I}$, our goal is to build a probabilistic model to predict the multivariate distribution of the translational vector $\mathbf{t} \in \mathbb{R}^3$ and rotational quaternion $\mathbf{q} \in \mathbb{R}^4$.

Denote the CNN feature extractor as $N(I_i, \theta_N)$. It takes RGB image I_i as input, and has learnable parameters θ_N. To model the uncertainty of prediction, we assume two independent Gaussian priors for $\{\mathbf{t}_i\}_{i=1}^n$ and $\{\mathbf{q}_i\}_{i=1}^n$, and consider the output from $N(I_i, \theta_N)$ as the shared input features for both of the SVI GP regressors. Based on the Bayesian knowledge in the previous section, the predictive distribution for translation component is

$$p(\mathbf{t}^*) = \int p(\mathbf{t}^*|\mathbf{u}) q_g(\mathbf{u}) d\mathbf{u}, \tag{4.18}$$

with

$$p(\mathbf{t}^*|\mathbf{u}) = \mathcal{N}(\mathbf{t}^*|\mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{m}_t, \mathbf{K}_{**} - \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*}), \tag{4.19}$$

and

$$q_g(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m}, \mathbf{S}). \tag{4.20}$$

The integral results in

$$p(\mathbf{t}^*) = \mathcal{N}(\mathbf{t}^*|\mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{m}_t, \mathbf{K}_{**} - \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*} + \mathbf{K}_{*m}\mathbf{K}_{mm}^{-1}\mathbf{S}_t^{-1}\mathbf{K}_{mm}^{-1}\mathbf{K}_{m*}). \tag{4.21}$$

$\mathbf{K}_{**} = K\left(N(I^*, \theta_N), N(I^*, \theta_N)\right)$ is the kernel matrix built on the features learnt from the CNN base. Since the priors and likelihoods are all Gaussian,

the predictive distribution for translation and quaternion are two multi-variate Gaussian. This distribution has learnable parameters $\mathbf{Z}_t$, $\mathbf{m}_t$, $\mathbf{S}_t$ and parameters for the kernel function θ_k. Similar results for rotation can be obtained by replacing the subscript t with q.

4.3.2 Coregionalization Kernel

Matrix $\mathbf{K}$ in the previous sections is the covariance of the function values in the Gaussian prior. It is built from the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ that describes the similarity between two points. For 1-d GPR, the output of k_{rbf} is a scalar and the kernel matrix for the whole training points $\mathbf{K}_{nn} = K(\mathbf{X}, \mathbf{X})$ is a $n \times n$ Positive Semi-definite (PSD) matrix. However in our task, both of the translation and quaternion are vector-valued functions, and have correlation between their entries. The scalar does not satisfy the need of storing this interrelationship. Alvarez et al. 2012 review the *cokriging* from geostatistics, and formally introduce the kernel for the vector-valued function GPs, the Coregionalization kernel. The key idea of coregionalization is to have a PSD $\mathbf{B} \in \mathbb{R}^{D \times D}$ act as a learnable parameter to represent the correlation between these functions, where D is the dimension of the output. To ensure the PSD property, a coregionalization kernel is built by

$$\mathbf{B} = \mathbf{W}\mathbf{W}^T + \text{diag}(\kappa), \tag{4.22}$$

where $\mathbf{W} \in \mathbb{R}^{D \times R}$ and $\kappa \in \mathbb{R}^D$. Both of them are learnable parameters during training. Size R is the rank of $\mathbf{B}$ (specified by the user at algorithm design time).

As a result, the full kernel matrix for inducing points $\mathbf{Z}$ in SVI is given as

$$\mathbf{K}^c_{mm} = K^c(\mathbf{Z}, \mathbf{Z}) = \mathbf{B} \otimes \mathbf{K}_{mm}, \tag{4.23}$$

where $\otimes$ is the Kronecker product of matrices. Now the size of full kernel matrix $\mathbf{K}^c_{mm}$ is $mD \times mD$.

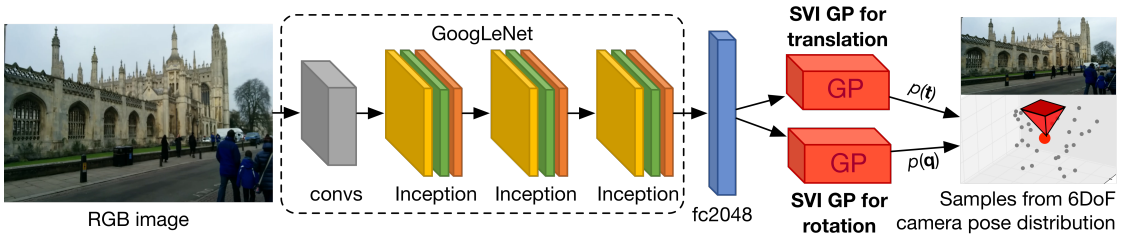


Figure 4.1: The overview of the GPoseNet. It takes a monocular RGB image as input. The high-level feature from *fc2048* layer of the CNN base is fed to two SVI GPs to perform probabilistic inference for translation and rotation. Our system outputs a distribution for camera relocalisation. The red dot and pyramid indicate the point estimate of the 6DoF pose.

4.3.3 Architecture

To make a fair comparison between linear regressors in (Kendall et al. 2016; Kendall et al. 2015) and the SVI GP regressors in our framework, we use the same deep feature encoder as PoseNet. We remove the last two fc layers of PoseNet and replace them with two SVI GP regressors. They take the output from the previous fc2048 layer as input, and perform Bayesian inference to obtain the distribution of translation and rotation for camera pose. The "loss function" of this framework is the sum of the objectives of two GPs, which will be further addressed in the next paragraph.

We use the coregionalization kernel in our system. Since the main goal of this work is to compare the capability of this hybrid architecture and the pure CNN structure, we did not tune the type of kernel function to seek for better performance. We use RBF as the base kernel function for simplicity, and leave the selection of kernel function to future work. Figure 4.1 illustrates the structure of our framework.

4.3.4 Objective Function

To train the whole structure in an end-to-end fashion, we design a multi-component objective function that combines CNN loss and the ELBOs of two log marginal likelihoods as follow:

$$L = \beta_{g_t}\mathcal{L}_{svi}(\mathbf{m}_t, \mathbf{S}_t, \mathbf{Z}_t) + \beta_{g_q}\mathcal{L}_{svi}(\mathbf{m}_q, \mathbf{S}_q, \mathbf{Z}_q) + \beta_{n_t}\left\|\hat{\mathbf{t}} - \mathbf{t}\right\|_2 + \beta_{n_q}\left\|\hat{\mathbf{q}} - \mathbf{q}\right\|_2, \quad (4.24)$$

where $\mathcal{L}_{svi}(\mathbf{m}_t, \mathbf{S}_t, \mathbf{Z}_t)$ is the ELBO of translation and $\mathcal{L}_{svi}(\mathbf{m}_q, \mathbf{S}_q, \mathbf{Z}_q)$ is the ELBO of quaternion. We will discuss the choices of β_{g_t} and β_{g_q} for these two ELBOs in the following section.

The last two components of equation (4.24) are used to learn the low-level and middle-level pose-related feature maps from the RGB images. In the original PoseNet, the final objective of the network consists of the pose losses produced by three pose regressors, which are placed after different levels (low-level, middle-level and high-level) of CNN feature maps. Only the last pose regressor at the output end is used at inference time to predict the pose. The first two regresses are added to learn pose-related shallow feature maps at training time, which are beneficial to the convergence of the network. In our system, we replace the final pose regressor with the proposed probabilistic objective, and keep the first two regressors. We use the same weights (β_{n_t} and β_{n_q}) for them as PoseNet (Kendall et al. 2015).

4.3.5 Hyperparameters

The main issue of the multi-task CNNs is that the norm-based losses for these targets are not always at the same unit and scale, therefore they need different weights to penalize. In the proposed objective function (4.24), however, the first two components are the objectives from the SVI GPs.

Maximizing the ELBO of $\log p(\mathbf{y})$ is mathematically equivalent to minimizing the KL divergence between the exact posterior $p(\mathbf{f}|\mathbf{y})$ and the variational distribution $q(\mathbf{f})$ (Titsias 2009). We observed that this measure between two distributions can reduce the dependence on the choice of hyperparameters.

For a clear understanding of this advantage, we use these two univariate Gaussians over a same random variable – $p_1(x) = \mathcal{N}(x|\mu_1, \sigma_1^2)$ and $p_2(x) = \mathcal{N}(x|\mu_2, \sigma_2^2)$ – as examples to simplify the proof. The KL divergence between

these two distributions is

$$\mathrm{KL}(p_1(x)||p_2(x)) = \log\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}. \tag{4.25}$$

It is straightforward to see that $(\mu_1 - \mu_2)^2$ and σ_2^2 have the same scale, because $\sigma_2^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_2)^2$ and x_i has the same scale with μ_1, as well as μ_2. The scale is canceled out in this equation. It means that the unit of the KL divergence is always 1, hence the choice of the hyperparameters β_{g_t} and β_{g_q} becomes easier. In the following experiments, we always keep them equal.

4.4 Experiments

4.4.1 Datasets

To benchmark our model both on outdoor and indoor scenarios both in this chapter and the following one, we use two datasets for training and evaluation, the Cambridge Landmarks (Kendall et al. 2016) and the 7Scenes (Shotton et al. 2013) dataset.

The dataset of Cambridge Landmarks is an outdoor urban localization dataset created by Kendall et al. 2016 using a hand-held camera in 5 different scenes. 3D models of these scenes are built from SfM (Wu 2013) algorithm. It was collected from many different points in time representing different lighting and weather conditions, as well as exhibits significant urban clutter such as pedestrians and vehicles.

7Scenes by Shotton et al. 2013 has 7 indoor scenes captured using a Kinect camera, provided with RGB-D images and ground truth poses. We only use the RGB images and ground truth poses to train our models. Note that the depth images can be very helpful to supervise the learning of the model, however our work focuses on the case that only relies on RGB input. Hence we omit the ground truth depth in our experiments.

		Geometry-based		Pure-RGB-based		
Scene	Spatial Extent(m)	Active Search (SIFT) (Sattler et al. 2016)	Geometric loss PoseNet (Kendall et al. 2017)	Bayesian PoseNet (Kendall et al. 2016)	PoseNet Spatial LSTM (Walch et al. 2017)	GPoseNet (Ours)
King's College	140×40	0.42m, 0.55°	0.88m, 1.04°	1.74m, 4.06°	**0.99**m, 3.65°	1.61m, **2.29**°
Old Hospital	50×40	0.44m, 1.01°	3.20m, 3.29°	2.57m, 5.14°	**1.51**m, 4.29°	2.62m, **3.89**°
Shop Facade	35×25	0.12m, 0.40°	0.88m, 3.78°	1.25m, 7.54°	1.18m, 7.44°	**1.14**m, **5.73**°
St Mary's Church	80×60	0.19m, 0.54°	1.57m, 3.32°	2.11m, 8.38°	**1.52**m, 6.68°	2.93m, **6.46**°
Chess	3×2×1	0.04m, 1.96°	0.13m, 4.48°	0.37m, 7.42°	0.24m, **5.77**°	**0.20**m, 7.11°
Fire	2.5×1×1	0.03m, 1.53°	0.27m, 11.3°	0.43m, 13.7°	**0.34**m, **11.9**°	0.38m, 12.3°
Heads	2×0.5×1	0.02m, 1.45°	0.17m, 13.0°	0.31m, **12.0**°	**0.21**m, 13.7°	**0.21**m, 13.8°
Office	2.5×2×1.5	0.09m, 3.61°	0.19m, 5.55°	0.48m, **8.04**°	0.30m, 8.08°	**0.28**m, 8.83°
Pumpkin	2.5×2×1	0.08m, 3.10°	0.26m, 4.75°	0.61m, 7.08°	**0.33**m, 7.00°	0.37m, **6.94**°
Red Kitchen	4×3×1.5	0.07m, 3.37°	0.23m, 5.35°	0.58m, **7.54**°	0.37m, 8.83°	**0.35**m, 8.15°
Stairs	2.5×2×1.5	0.03m, 2.22°	0.35m, 12.4°	0.48m, 13.1°	0.40m, 13.7°	**0.37**m, **12.5**°

Table 4.1: Median error of localization for Cambridge Landmarks and 7Scenes datasets. We compare our method (GPoseNet) with the Spatial LSTM-PosNet (Walch et al. 2017), Bayesian PoseNet (Kendall et al. 2016). For Cambridge Landmarks dataset and 7Scenes datasets, the median pose error of our method is averagely ($2.0m$, 4.6°) and ($0.3m$, 9.9°). The overall results surpass Bayesian PoseNet and are comparable with Spatial LSTM-PoseNet Walch et al. 2017, for which the average of median error for these two datasets are (1.3m, 5.5°) and (0.3m, 9.9°) respectively. The best performance is highlighted in bold only for Pure-RGB-based methods.

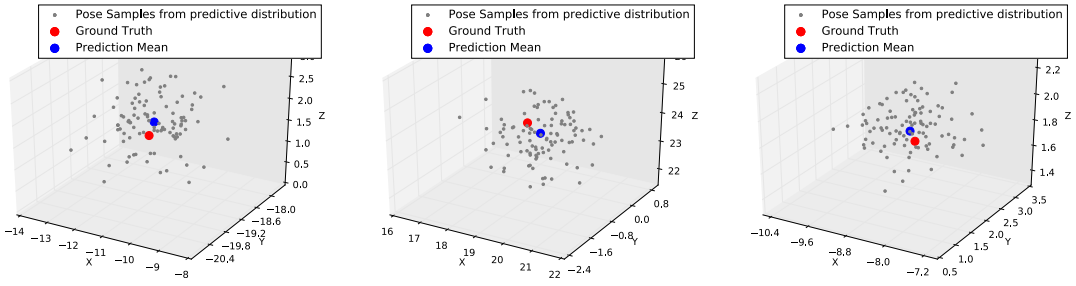
4.4.2 Training Regime

We follow the same training/test split in PoseNet. All the experiments are done on a NVIDIA GeForce GTX 1070 GPU. The batch size in training is 75. We optimize all the models in an end-to-end fashion with ADAM (Kingma et al. 2014). We also initialize the CNN base with pre-trained weights from ImageNet (Deng et al. 2009), suggested by Kendall et al. 2015 and Kendall et al. 2016. The number of inducing points for SVI GPs is 10% of the image number in each training split. This number varies w.r.t different scenes, from to 23 to 149 in Cambridge Landmarks dataset. We initialize the inducing points $\mathbf{Z}$ with the results from k-means clustering over the features from 500 images. These images are randomly selected from training set. This initialization keeps the induced feature points and the deep features of the training images in the same domain, preventing the large – and meaningless – elements in the kernel matrix, which could raise if $\mathbf{Z}$ is with random initialization around zero. Experiments show that this initialization also ensures a stable convergence. The learning rate is 10^{-4} for CNN base and 10^{-2} for GPs' parameters. We implement the CNN base with Tensorflow (Abadi et al. 2016) and the GPs part with GPflow (De G. Matthews et al. 2017).

We evaluate our method from two perspectives, localization accuracy and predictive uncertainty. Overall, the results from the following experiments shows that our method can achieve comparable accuracy with the state-of-the-art pure RGB based method, Spatial LSTM PoseNet (Walch et al. 2017), and estimates the uncertainty in a more efficient way comparing to Bayesian PoseNet (Kendall et al. 2016).

4.4.3 Localization Accuracy

We use the mean of the translational and rotational predictive distributions as the point estimation for the 6DoF camera pose. The rotational vector is normalized to ensure that it is an unit vector. In table 4.1, we compare the median error of localization in different scenes with the state-of-the-art methods. We can see that with



(a) King's College (b) Old Hospital (c) Shop Facade

Figure 4.2: Position samples from the predictive distribution. We show 100 samples from three predictive pose distributions of our models from Cambridge Landmarks dataset.

the same CNN encoder, our SVI GP regressors outperform PoseNet and Bayesian PoseNet in every scene, and have similar results with Spatial LSTM PoseNet Walch et al. 2017. In the scene *Old Hospital* and *Stairs*, the proposed method produces similar accuracy with geometry-based method (Kendall et al. 2017).

This result shows that by replacing the L2-loss based pose regressors with the SVI GPs, our system improves the performance of the original PoseNet and Bayesian PoseNet. Since all of them use same CNN base, hence the advancement is contributed by the regressors. The comparable accuracy with Spatial LSTM PoseNet suggests that the effect of regressors replacement qualitatively equals the enhancement of the output feature from CNN.

4.4.4 Uncertainty Evaluation

The "cherry on top" of this proposed framework is the ability to predict a distribution over the 6DoF pose without losing the localization accuracy. Figure 4.2 shows the samples from estimated position distribution for three test images in different scenes.

First, we compare the distributions of our method and Bayesian PoseNet in terms of efficiency. The pose distribution of Bayesian PoseNet is summarized from the Monte Carlo samples. The number of samples is also the number of inference times for one image. The more poses sampled, the more time consumed. Parallelization

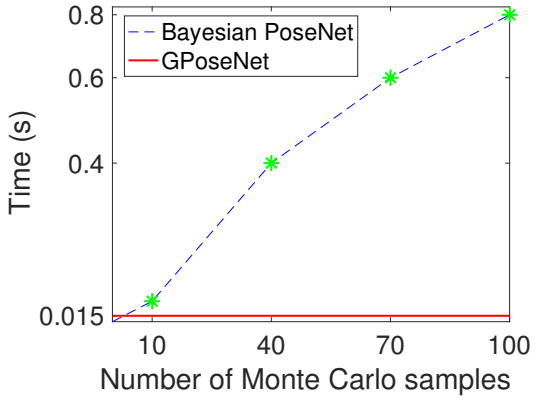


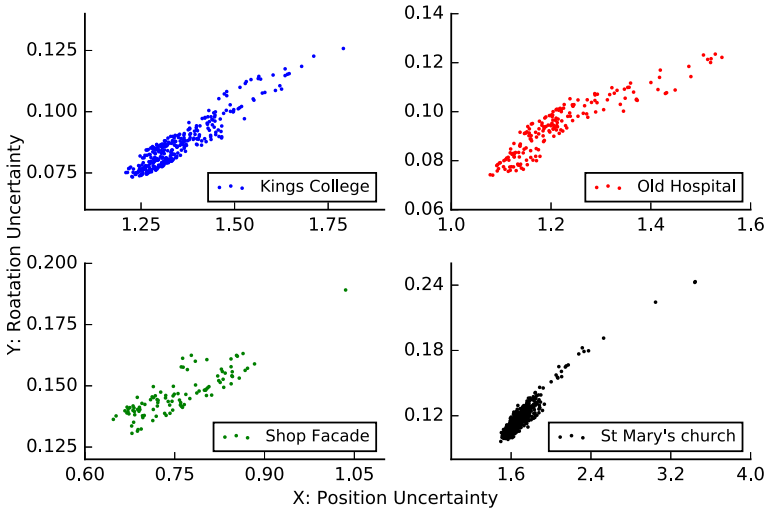
Figure 4.3: Comparison of system efficiency. For Bayesian PoseNet, the average time consumption for probabilistic inference is correlated to the number of Monte Carlo samples.

might help sampling method but this is still an issue when the computational resource is limited such as using only CPU for inference. In figure 4.3, we plot the average time for pose distribution estimation of one image against the number of samples[1]. If the number of samples is 40, it takes 0.4 second to estimate the pose distribution in average.

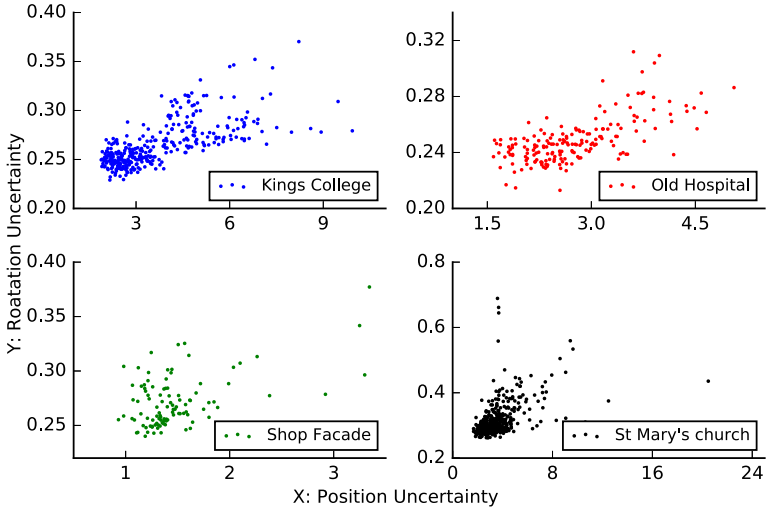
In contrast, the number of inference in our method for pose distribution is only one. As shown in figure 4.3, it takes 0.015 second to perform the distribution prediction, which is lower than the Bayesian PoseNet when the sample number is 10. Since the distribution is not from the sampling method, the time consumption is not related to the number of samples. This significant improvement of efficiency makes our system ready for real time relocalisation with uncertainty.

To qualitatively evaluate the rotational and traditional uncertainty of our model, we use the same measure in Bayesian PoseNet (Kendall et al. 2016), which is the trace of covariance matrix for each predicted pose component. In the following evaluation, the term *uncertainty* stands for this trace. In (Kendall et al. 2016), the authors have found the trace to be an effective scalar measure of uncertainty. The

[1] Due to the performance of our GPU, the time consumption of Bayesian PoseNet inference in this chapter is more than Bayesian PoseNet (Kendall et al. 2016). However we perform all the experiments using the same hardware to ensure a fair comparison.



(a) GPoseNet



(b) Bayesian PoseNet

Figure 4.4: The correlation between uncertainty of translation and rotation. This shows that the translation uncertainty is linearly correlated with rotation uncertainty, and the linearity is more obvious in our distribution compared to Bayesian PoseNet.

trace is a sum of the eigenvalues, which is rotationally invariant and represents the uncertainty that the Gaussian contains effectively. This form of uncertainty measure is strongly correlated with metric error in translation and rotation, which shows that we can use the uncertainty estimate to predict relocalisation error.

The translational uncertainty and rotational uncertainty from our model are more strongly linear correlated than Bayesian PoseNet (Kendall et al. 2016). In figure 4.4, we show these two uncertainties (traces of covariance matrices) from

	King's College	Old Hospital	Shop Facade	St Mary's Church
King's College	78%	20%	1%	1%
Old Hospital	19%	61%	17%	3%
Shop Facade	0%	37%	63%	0%
St Mary's Church	13%	14%	2%	71%

(a) GPoseNet

	King's College	Old Hospital	Shop Facade	St Mary's Church
King's College	75%	6%	12%	7%
Old Hospital	12%	80%	8%	0%
Shop Facade	5%	14%	76%	5%
St Mary's Church	2.5%	5%	3%	79.5%

(b) Bayesian PoseNet

Figure 4.5: Confusion matrices of model uncertainty. The test images from each dataset (row) are tested on the each model (column). We consider the model that the lowest uncertainty belongs to as the classified scene. To be more specific, 78% (row 1 column 1 of figure (a)) means that 78% of the test images in King's College set achieve the lowest Z-score on the model trained from King's College set, and they are correctly classified as image in King's College set, which are true positives. Whereas 20% (row 1 column 2 of figure (a)) means that 20% of the test images in King's College set achieve the lowest Z-score on the model trained from Old Hospital set (they are therefore classified as images from Old Hospital, which are false negatives). 19% (row 2 column 1 of figure (a)) means that 19% of the test images in Old Hospital set achieve the lowest Z-score on the model trained from King's College set, which are false positives.

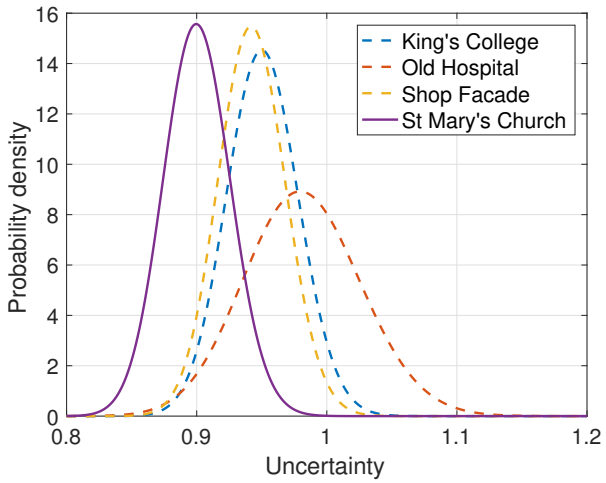
scenes in Cambridge Landmarks datasets. This linear correlation is in accordance with the results from Bayesian PoseNet, but with a more consistent linearity.

We also explore various of applications of the estimated uncertainty in the task of pose estimation. The uncertainty of camera pose can be interpreted as a measure for scene classifier. Kendall et al. 2016 define a normalized metric, Z-score, to compare the uncertainties of different models. To compute the Z-score for a model, they firstly test all images from the scene that the model was trained on are inferred by the model, and a Gamma distribution is fit over the uncertainties of the test results. Then, when a *new* image (from scene's test images split or images from other scene) is inferred by the model, its uncertainty is sit in the Gamma distribution and compared to the population. The percentile of the *new* uncertainty

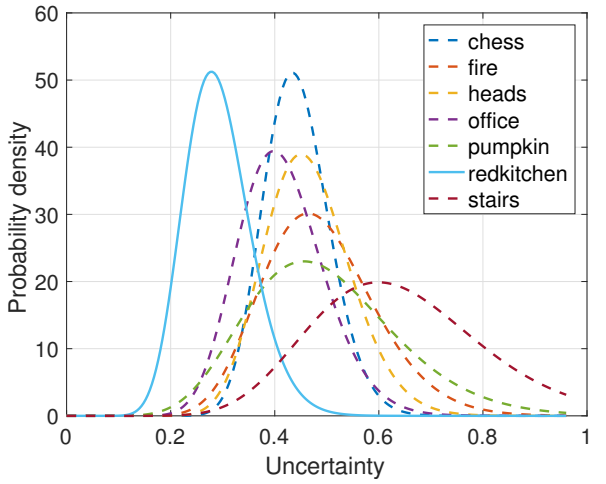
in the Gamma distribution is defined as the Z-score for the image. We evaluate the test split from one scene on all models. For example, one test image from Cambridge Landmarks dataset has four Z-scores: on the trained model from scene King's College, Old Hospital, Shop Facade, St Mary's Church, respectively. This test image is classified to one scene if the Z-score inferred by that model is lowest. Figure 4.5 compares the confusion matrices of our method and Kendall et al. 2016. It shows that the smallest predictive uncertainty of a test image is majorly produced by the model that is trained on the analogous training split.

Secondly, the uncertainty of our model also indicates the confidence of the pose prediction. The confusion matrices in figure 4.5 is from a setting we call 'same-image-for-different-models'. This means the comparison is done by applying different models on a same image, and then compare the it Z-scores. We alternatively run an experiment on the 'different-images-for-same-model' setting, which is an intuitive evaluation scheme for each individual model. To to so, we evaluate the test splits from all scenes on one of the models, and plot the probability density function of the approximated Gamma distribution of uncertainties in figure 4.6. We can see that if the images are from the test split of the dataset that this model is trained on, the model tends to estimate a lower uncertainty. This observation corroborates the conclusion of the vanilla GPR.

It means that the uncertainty from our model is a practicable indicator for the confidence of the inferred result. We suggest that this confidence also can be used in other tasks beyond pose regression, such as image classification or object detection, for which the most of the deterministic CNNs trust the prediction with absolute certainty.



(a) St Mary's Church



(b) Red Kitchen

Figure 4.6: The gamma distribution of uncertainties on chosen model. We evaluate all the test images from four scenes on the model of scene *St Mary's Church* and *Red Kitchen* to obtain uncertainties. For each test set from all four scenes, we plot the approximated Gamma distributions of the uncertainties. This shows that these two models produce smaller uncertainties on the test images from the corresponding scenes, which means the model from St Mary's Church is more confident about the pose for a test image from scene St Mary's Church, and vice versa.

4.5 Conclusion

In this chapter, we show how to combine the deterministic CNN and probabilistic GPR together to accomplish real time camera relocalisation with modelling the uncertainty. This is done by replacing the traditional L2 norm loss based linear regressor with KL divergence based SVI GPs regressor. It improves the system

efficiency of method based on bayesian approximate CNN without losing accuracy.

However, compare to the traditional methods which use geometric rules to derive the 6dof pose, the performance of PoseNet and its variants (including the proposed method in this chapter) is still far from being accurate for real world use. In next chapter, we will address this problem for camera relocalisation by using the idea of scene coordinates, and deeply integrate rules in geometry to the learning-based pose estimation to explore space of improvement in accuracy.

Bibliography

Kendall, Alex, Matthew Grimes, and Roberto Cipolla (2015). "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946.

Davison, Andrew J, Ian D Reid, Nicholas D Molton, and Olivier Stasse (2007). "MonoSLAM: Real-time single camera SLAM". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 6, pp. 1052–1067.

Kendall, Alex and Roberto Cipolla (2016). "Modelling uncertainty in deep learning for camera relocalization". In: *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, pp. 4762–4769.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*, pp. 1050–1059.

Williams, Christopher KI and Carl Edward Rasmussen (2006). *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA.

Hensman, James, Nicolo Fusi, and Neil D Lawrence (2013). "Gaussian processes for big data". In: *arXiv preprint arXiv:1309.6835*.

Kendall, Alex and Roberto Cipolla (2017). "Geometric loss functions for camera pose regression with deep learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5974–5983.

Damianou, Andreas and Neil Lawrence (2013). "Deep gaussian processes". In: *Artificial Intelligence and Statistics*, pp. 207–215.

Csató, Lehel and Manfred Opper (2002). "Sparse on-line Gaussian processes". In: *Neural computation* 14.3, pp. 641–668.

Snelson, Edward and Zoubin Ghahramani (2006). "Sparse Gaussian processes using pseudo-inputs". In: *Advances in neural information processing systems*, pp. 1257–1264.

Quiñonero-Candela, Joaquin and Carl Edward Rasmussen (2005). "A unifying view of sparse approximate Gaussian process regression". In: *Journal of Machine Learning Research* 6.Dec, pp. 1939–1959.

Titsias, Michalis (2009). "Variational learning of inducing variables in sparse Gaussian processes". In: *Artificial Intelligence and Statistics*, pp. 567–574.

Alvarez, Mauricio A, Lorenzo Rosasco, Neil D Lawrence, et al. (2012). "Kernels for vector-valued functions: A review". In: *Foundations and Trends in Machine Learning* 4.3, pp. 195–266.

Sattler, Torsten, Bastian Leibe, and Leif Kobbelt (2016). "Efficient & effective prioritized matching for large-scale image-based localization". In: *IEEE transactions on pattern analysis and machine intelligence* 39.9, pp. 1744–1756.

Walch, Florian, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers (2017). "Image-based localization using lstms for structured feature correlation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 627–637.

Shotton, Jamie, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon (2013). "Scene coordinate regression forests for camera relocalization in RGB-D images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937.

Wu, Changchang (2013). "Towards linear-time incremental structure from motion". In: *2013 International Conference on 3D Vision-3DV 2013*. IEEE, pp. 127–134.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE, pp. 248–255.

Abadi, Martn, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. (2016). "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". In: *arXiv preprint arXiv:1603.04467*.

De G. Matthews, Alexander G, Mark Van Der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman (2017). "GPflow: A Gaussian process library using TensorFlow". In: *The Journal of Machine Learning Research* 18.1, pp. 1299–1304.

5

Camera Relocalisation by Exploiting Multi-View Constraints for Scene Coordinates Regression

Contents

This chapter introduces a method that estimates the 6dof camera pose using scene coordinate regression from RGB image. Existing methods require the ground truth annotations for scene coordinate learning. We propose to use multi-view

geometry to provide indirect supervision. The fundamental idea is to use an image-based warp error between different views of a scene point to improve the ability of the network to regress to the correct absolute scene coordinates of the point. It is further augmented with a featuremetric error. This work was published in the workshop on Deep Learning for Visual SLAM of IEEE International Conference on Computer Vision 2019.

5.1 Introduction

The common scenarios where the camera relocalisation module is applied, always require high accuracy in the 6dof pose, because the results from relocalisation have to be mathematically good enough to perform correction over the drift caused by a pose tracker. Though direct pose regression using deep network is an interesting direction to explore with advantages such as being robust to a variety of dynamic effects in the images including lighting changes and motion blur, however in those demanding applications, its performance is currently not sufficiently accurate. Sattler et al. 2019 have proved that pose regression is more closely related to pose approximation via image retrieval than to accurate pose estimation via 3D structure. This means that the generalization beyond the training data to novel poses is not guaranteed, and the most promising solution to the improvement is to bring the 3D model into the system.

In order to integrate the 3D structure to the learning-based method for pose estimation, Shotton et al. 2013 introduce the concept of scene coordinates. They are basically defined as the 3D coordinates of the points in the scene-centric (or world-centric) frame. Because of having no dependency on camera positions, the representation of scene coordinates is globally consistent for a visual feature no matter what the viewpoint is. Leveraging from this consistency, work such as (Shotton et al. 2013; Brachmann et al. 2016; Brachmann et al. 2017; Brachmann et al. 2018; Bui et al. 2018) aim to build learning models that map the visual

appearances on an image (or combined with depth information) to scene coordinates. In such a way, the 2D-3D correspondence can be built via learning algorithms, which can then be used to determine the pose in the manner of classical solvers (Gao et al. 2003; Lepetit et al. 2009; Hesch et al. 2011; Wang et al. 2018).

CNNs have also been studied to serve as the base for the learning model. As successful representatives, DSAC++ (Brachmann et al. 2018) and its predecessor DSAC (Brachmann et al. 2017) apply a FCN (Long et al. 2015) over the image to perform dense scene coordinate regression for every pixel. In that sense, these models can also be interpreted as a network that performs 3D scene reconstruction from a single image at inference time, and then estimate the pose as a post process. Because of the robustness of the CNN coordinate regressor and the strong geometric constraints of Perspective-n-Point (PnP) solver, this line of methods achieves great accuracy for pose estimation, and even outperforms the traditional approaches Sattler et al. 2016.

The key to these methods working well is the ability of the deep model to map to the fixed 3D location of any given scene point from an image of that point. Since the viewpoint can be anywhere, the appearance of the scene point may vary, but the network should still regress to the same global 3D coordinates. It is not clear if such a network is capturing the invariance of features to different viewpoints and therefore implicitly encoding multi-view geometric constraints (Hartley et al. 2003), or if it is acting as a huge look-up table that simply memorizes all possible appearances and corresponding mappings. Regardless, in this chapter we aim to make the multi-view constraints more explicit during training.

To that end, the main innovation of this chapter is to exploit constraints from multi-view geometry to supervise the learning of a model for scene coordinate regression. We aim to retain the advantages of the training for single view reconstruction, but to incorporate the additional information available from viewpoint invariant image features under motion parallax. Specifically, after predicting the

scene coordinates from one image in the database during training, we project the predictions to another image that shares an intersection of the camera frustum with the query image, using the ground truth pose of the target image. We then compare local image feature descriptors – any difference that we assume arises from an error in the predicted scene coordinates – and use this error for back-propagation. In this chapter we explore two types of local image features: (i) simple RGB values (which are invariant to viewpoint under the common lambertian reflection assumption); (ii) high dimensional features that are learned to be good for matching (Weerasekera et al. 2017; Zhan et al. 2018).

The advantage of our method is that it produces more accurate scene coordinates compared to the single-view training approaches. Therefore it yields better 2D-3D correspondence for single view pose estimation using RANSAC during test. On top of this accuracy improvement, our system also avoids the scale issue that the methods with single view training may suffer. The reason for the first stage – training with pseudo depth in the RGB-only case (See section 5.2 for details) – is needed in DSAC++ (Brachmann et al. 2018) is that it assigns an initial scale to the scene coordinates. A good guess of the scale helps the next learning stages and vice versa. This makes it heavily reliant on the heuristic. In contrast, experiments show that our method relaxes the requirement for this strong prior through the use of multi-view geometry. One should bear in mind that our training pipeline also requires the initialization stage, but only a rough guess for depth is needed to avoid the case when all the photometric/feature construction losses are meaningless.

A similar technique has been applied to the topic of self-supervised depth estimation (Garg et al. 2016; Zhan et al. 2018; Zhou et al. 2017). The differences between these works and ours are twofold. First, the objectives are different. Depth estimation focuses on purely recovering the geometric structure of each frame. However in our task, the scene coordinates inferred from the network are intermediate values whose purpose is ultimately to enable camera pose estimation.

Second, the *label consistency* of these two representations is different, and this has a significant effect on learning. In the case of depth estimation, the labels are the depth values of each pixel. As the camera moves these depth values change even for the same part of the scene because they are camera position dependent. In contrast, for the scene coordinate estimation task, the scene coordinates are described in a fixed world coordinate frame – the label for a scene point is its 3D coordinates and this label is consistent across all viewing locations and appearances, *i.e.*, independent of camera location. We believe this makes the 3D scene coordinates easier to regress than depth values in the known environments.

5.2 Training in DSAC and DSAC++

Please refer to chapter 2 and section 4.2.1 for the details of work related to camera relocalisation. We mainly review the training of DSAC (Brachmann et al. 2017) and DSAC++ (Brachmann et al. 2018) which is strongly related to our work and will be referred frequently in this chapter.

A multi-step scheme is adopted in the training phase of DSAC++ (Brachmann et al. 2018). The training of the scene coordinate regression CNN consists of three stages and the performance of the model progressively increases with additional training. Since they will be repetitively mentioned hereinafter, we give a brief introduction to them. The scene coordinate regression model is initially trained with either ground truth scene coordinates or a heuristic assuming a constant distance of the scene, depending on the availability of depth images or the 3D scene model. In the second stage, the model is enhanced by the supervision from the distance between the 2D projection of predicted scene coordinates (given the ground truth camera pose) and the ideal image pixel position, namely the reprojection error. In the third step, DSAC++ (Brachmann et al. 2018) refines the model with an end-to-end scheme that combines the inlier soft counting based hypotheses scoring and

differentiable refinement that is mentioned above, resulting in superior performance.

5.3 Method

The overall objective of this chapter is to efficiently train an FCN-based scene coordinate regression model using multi-view geometric constraints, and then apply this model to a single view RGB image to infer dense 2D-3D correspondences for pose estimation in a RANSAC pipeline. We start by describing the network architecture in section 5.3.1. In addition to the single view reprojection loss, we introduce three more supervisions that come from the multi-view geometry induced by camera motion. The photometric warp error based image reconstruction loss is introduced in section 5.3.2. An additional deep feature reconstruction loss which takes contextual information into consideration rather than per pixel colour alone is introduced in section 5.3.3. We propose a smoothness prior in 3D space to regularize training in section 5.3.4, in order to mitigate the effect of featureless, ambiguous-to-match scene regions. Figure 5.1 shows our framework in the training phase. The overall training loss and inference procedure are summarized in section 5.3.5 and section 5.3.6, respectively.

5.3.1 Scene Coordinate Regression

The FCN (Long et al. 2015) model of DSAC++ (Brachmann et al. 2018) is inherited into our system for scene coordinate regression. We denote this model as $\mathbf{w}$. The output of this network is the scene coordinate map $\mathbf{Y}(\mathbf{w}, I)$ of an input image I. Every element of this map is a 3D vector $\mathbf{y}_{i,j} \in \mathbb{R}^3$, which represents the coordinates in the world reference frame of the point that corresponds to an image pixel. This FCN comprises 12 convolutional layers, 3 of which have stride size 2. Thus, $\mathbf{Y}(\mathbf{w}, I)$ is one-eighth the size of the input image I. This means that the 3D scene coordinates predicted by the model represent that of the center of 8×8 pixel tiles in I. Note that however each output corresponds to an overall receptive field of 41×41 pixels.

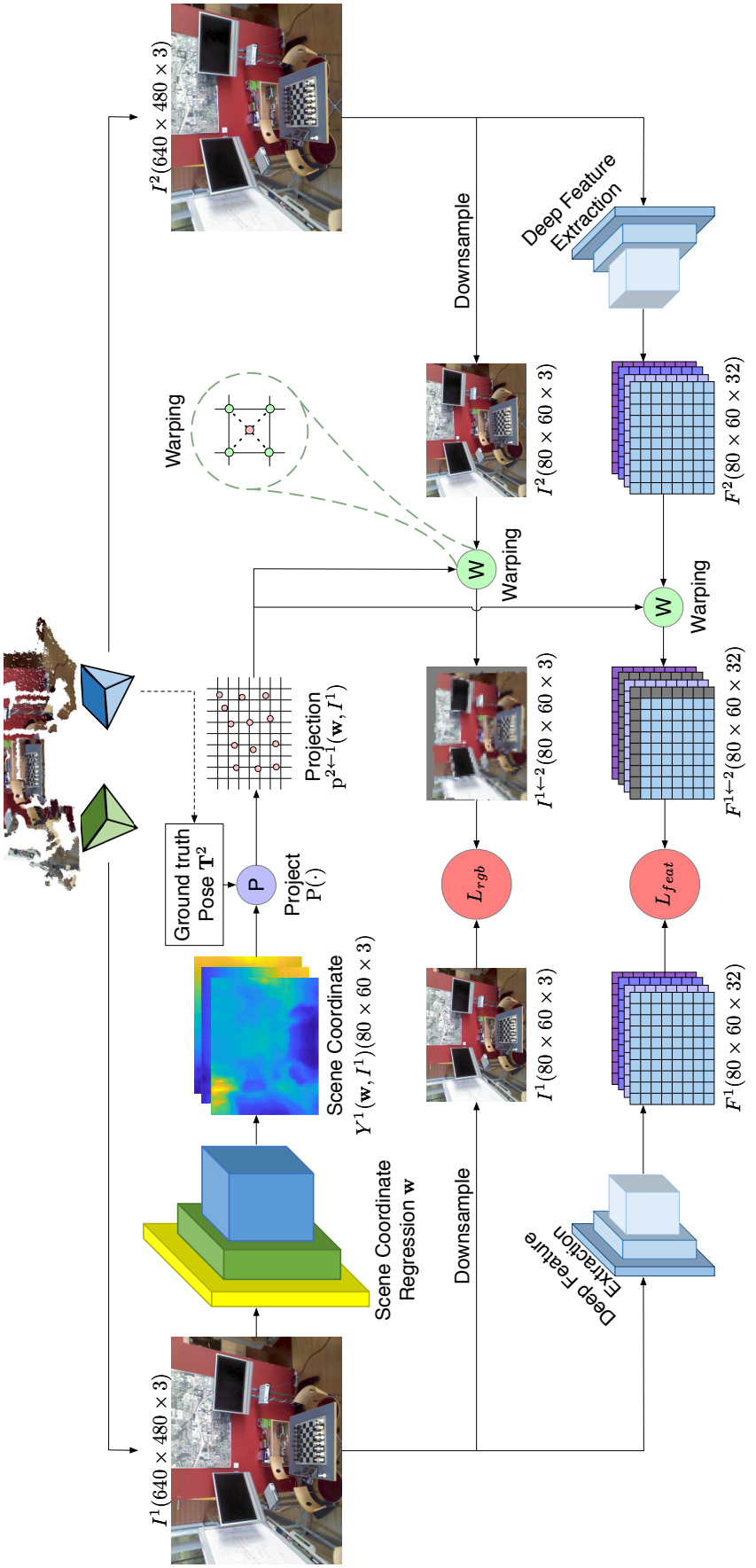


Figure 5.1: The training pipeline of our framework with photometric loss and feature reconstruction loss. The spatial size of all variables are specific for 7Scenes dataset. The reprojection loss and smoothness prior loss are omitted for simplicity.

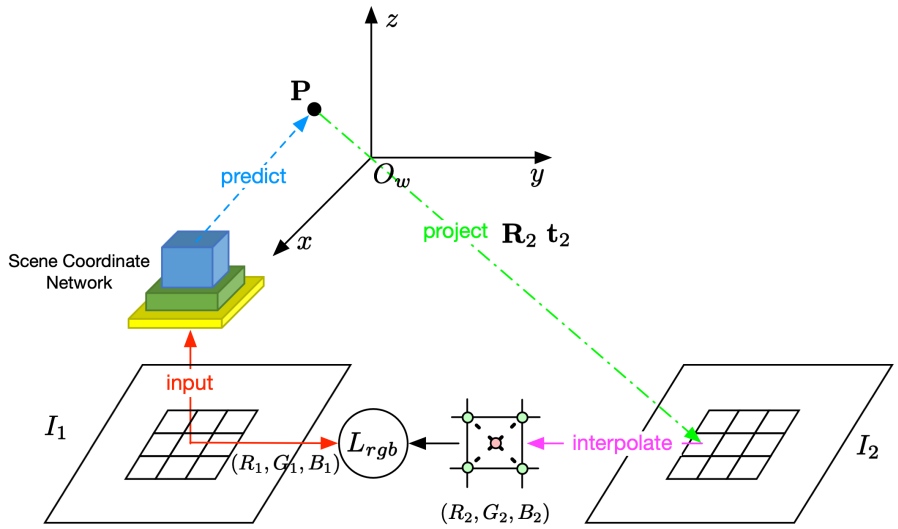


Figure 5.2: The build of photometric loss. The scene coordinate network predicts a 3D point P in world coordinate system for a pixel on the first image. The prediction is projected second image with ground truth pose $\mathbf{R}_2, \mathbf{t}_2$. The RGB value of the projection is computed using bilinear interpolation. The photometric loss is the distance between value of the input pixel and interpolated value. The featuremetric loss is computed in a similar way.

5.3.2 Photometric Reconstruction

The main supervision for learning the scene coordinate regression model in our framework comes from the reconstruction of two types of features: (i) RGB colour; (ii) deep features trained to be good for matching (Weerasekera et al. 2017) (used as an "off-the-shelf" tool). We begin by introducing the photometric (RGB) constraint in this section.

Given an image pair $\{I^1, I^2\}$ with known ground truth absolute poses $\mathbf{T}^1$ and $\mathbf{T}^2$, firstly I^1 is fed into the regressor $\mathbf{w}$ for predicting the map of the scene coordinates $\mathbf{Y}^1(\mathbf{w}, I^1)$. Then, they are projected onto the image plane of I^2 with the ground truth pose of second frame $\mathbf{T}^2$ and camera intrinsics $\mathbf{K}$, for computing the projected pixel positions $\mathbf{p}^{2\leftarrow1}(\mathbf{w}, I^1)$. Using the RGB values of I^2 at $\mathbf{p}^{2\leftarrow1}(\mathbf{w}, I^1)$, a warped image $I^{1\leftarrow2}$ is formed to synthesize image I^1. See figure 5.2 The procedure can be formulated as equation (5.1),

$$I^{1\leftarrow2} = f(\mathbf{Y}^1(\mathbf{w}, I^1), I^2, \mathbf{T}^2, \mathbf{K}), \tag{5.1}$$

where the function $f(\cdot)$ is the reconstruction function based on image warping. This operation is fully differentiable when using the bilinear interpolation reconstruction method proposed in Spatial Transformer Networks (STN) (Jaderberg et al. 2015), which guarantees differentiability in the whole system.

The loss based on photometric difference between the real image I^1 and the synthetic image $I^{1\leftarrow 2}$ is defined as

$$L_{rgb} = \frac{1}{H \times W} \sum_{m,n}^{H,W} \left\| I^1_{m,n} - I^{1\leftarrow 2}_{m,n} \right\|_1 , \tag{5.2}$$

where H and W are the spatial dimensions of the output scene coordinate map $\mathbf{Y}^1(\mathbf{w}, I^1)$.

5.3.3 Dense Deep Feature Reconstruction

Since the RGB values of an image are sensitive to change in the lighting condition, the consistency of the light/colour intensity of a 3D point across two images cannot always be assured, especially in uncontrolled environments. There are also cases in both indoor and outdoor scenes where a large patch of the image is filled with same RGB value due to lack of texture on the objects and surfaces in the scene. Photometric reconstruction loss is only useful in regions where intensity gradient is large. Hence, a robust dense image feature, which contains more contextual information, can be used for dealing with these issues. In this chapter, we exploit the deep CNN features for dense matching proposed by Weerasekera et al. 2017.

While any dense visual descriptor such as dense SIFT may be suitable for the dense matching task, the learned deep visual descriptor in Weerasekera et al. 2017 is light-weight allowing for efficient training, and has been proven to be successful for dense monocular reconstruction. To extract the deep features for each pixel in the image, the whole image is passed into a fully convolutional neural network which is pre-trained using the method in Weerasekera et al. 2017 on the raw NYU-D v2 dataset Silberman et al. 2012. A 32-dimensional feature map F with the same

spatial dimensions as the input image is regressed by the network which can be subsequently used for dense image alignment. We then downsize it to one-eighth of the image size to match the scene coordinate map. Given the feature map F^2 regressed for I^2, we can warp it into I^1's frame of reference as follows,

$$F^{1\leftarrow 2} = f(\mathbf{Y}^1(\mathbf{w}, I^1), F^2, \mathbf{T}^2, \mathbf{K}). \tag{5.3}$$

Similar to equation (5.2), the deep dense feature reconstruction loss is defined as

$$L_{feat} = \frac{1}{H \times W} \sum_{m,n}^{H,W} \left\| F^1_{m,n} - F^{1\leftarrow 2}_{m,n} \right\|_1 . \tag{5.4}$$

5.3.4 3D Smoothness Prior

The predicted scene coordinates from a single view image can be considered as the reconstruction of the scene. So far, the learning of our model for coordinate prediction only considers the input(image)-output(3D points) relationship. The correlation between the predicted 3D points is also important to recover the geometry of the scene. In particular, we utilize the intensity consistency within the image to constrain a smooth prediction in the coordinate map. A similar idea has been applied in (Garg et al. 2016; Godard et al. 2017; Heise et al. 2013; Zhan et al. 2018) in the depth estimation topic. We extend this mechanism to the 3D space.

The idea behind this smoothness prior is that a large 3D Euclidean distance between predicted neighboring scene coordinates should be penalized if there is no image evidence to support this (*e.g.* if the image is uniform). Specifically, it is formulated as

$$L_s = \sum_{m,n}^{H,W} e^{-|\partial_x I_{m,n}|} \left\| \partial_x \mathbf{Y}_{m,n} \right\|_2 + e^{-|\partial_y I_{m,n}|} \left\| \partial_y \mathbf{Y}_{m,n} \right\|_2 , \tag{5.5}$$

where $\mathbf{Y}$ is the predicted coordinate map, $\partial_x(\cdot)$ and $\partial_y(\cdot)$ are the horizontal and vertical gradient operators.

5.3.5 Training Loss

Apart from the three losses previously mentioned, we also use the single view reprojection error of I_1 as the base loss to train our model, since the ground truth pose $\mathbf{T}^1$ is available. The reprojection error loss is defined as

$$L_{repro} = \frac{1}{H \times W} \sum_{m,n}^{H,W} \left\| P(\mathbf{Y}^1, \mathbf{T}^1, \mathbf{K}) - \mathbf{p}^1 \right\|_2, \tag{5.6}$$

where $P(\cdot)$ is the projection function that projects a 3D point and computes its pixel position in the image plane. Note that this is the loss that DSAC++ used in the training of the second stage of their system.

Hence, the total loss that we use to train our model is

$$L = w_r L_{repro} + w_p L_{rgb} + w_f L_{feat} + w_s L_s, \tag{5.7}$$

where w_r, w_p, w_f and w_r are the loss weights hyper-parameters.

5.3.6 Single View Inference

Although our system is trained with image pairs, it only requires a single view image to perform inference. Once the model for scene coordinate prediction is trained, we can establish the dense correspondences between image pixel positions and the 3D points and then use RANSAC+PnP to estimate the pose of the camera.

Similar to DSAC++ (Brachmann et al. 2018), we first sample N sets of four 2D-3D correspondences using the predicted coordinate map (i.e. each sample contains four image points and corresponding 3D scene coordinates). After solving the PnP problems independently, a pool of N pose hypotheses is built for the best candidate selection. To rank the hypotheses, we compute the reprojection error map for each hypothesis using all predicted 3D coordinates. The best hypothesis is selected depending on the number of inliers, which is defined as the points whose reprojection error is less than a threshold τ. Finally, the best hypothesis is refined with updated inliers iteratively to produce the final pose estimate.

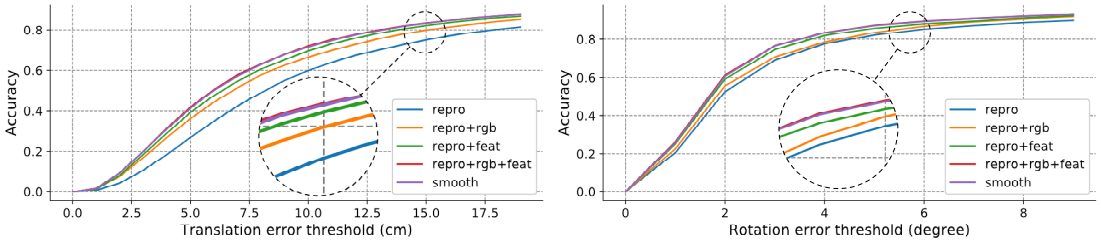


Figure 5.3: Localization accuracy of position and orientation as a cumulated histogram of errors. The horizontal axis is the threshold for transnational error (left, in cm) and rotational error (right, in degree). The vertical axis is the proportion of the test images of which transnational or rotational error is smaller than the thresholds on the horizontal axis.

5.4 Experiments

5.4.1 Data Preparation

Training images are selected following the official split of datasets 7Scenes (Shotton et al. 2013) and Cambridge Landmarks (Kendall et al. 2015) mentioned in section 4.4.1. Since the images are taken from a monocular camera, it is important to select proper target frames for each training image to enable multi-view geometry based supervision. To that end, we randomly select 3 images from its nearest [-100, +100] neighbours as the pair candidates (thanks to the fact that the images are from a continuous sequence). Then we use an off-the-self optical flow estimation method (FlowNet2.0 (Ilg et al. 2017) and its implementation (Reda et al. 2017)) to compute the overlap between the current frame and its pair candidates. We choose the candidate as the final pair image if the ratio of their overlap area to the image spatial area is within the range of [0.4, 0.9]. On average, a training image has ∼2 pairs to build multi-view constraints. We also use the overlap as the mask to zero out the meaningless reconstruction loss caused by the pixels that are projected out of frame on the target images.

5.4.2 Training and Test Regime

We use a two-stage scheme for our training pipeline. Firstly, we train the model with the heuristic suggested in DSAC++ (Brachmann et al. 2018) since only RGB

	repro(baseline)	repro+rgb	repro+feat	repro+rgb+feat	w/ smooth
Chess	5.03cm, 1.36° 49.6%	4.58cm, 1.56° 54.7%	3.63cm, **1.21°** 70.75%	**3.59cm**, 1.23° **71.1%**	**3.59cm**, 1.23° 69.9%
Fire	9.41cm, 3.00° 26.8%	7.94cm, 2.99° 38.9%	5.28cm, 1.92° 47.9%	**5.05cm**, **1.87°** **49.75%**	5.32cm, 2.04° 48.25%
Heads	24.9cm, 10.0° 6.4%	6.05cm, 3.67° 46.5%	13.2cm, 7.45° 21.1%	5.02cm, 3.22° 48.9%	**4.99cm**, **2.98°** **50.1%**
Office	6.21cm, 1.43° 36.25%	6.00cm, 1.48° 39.28%	5.71cm, 1.35° 42.18%	**5.62cm**, **1.34°** **42.78%**	5.71cm, 1.36° 41.83%
Pumpkin	7.26cm, 1.77° 32%	6.23cm, 1.60° 36.8%	5.60cm, 1.48° 44.33%	**5.56cm**, **1.47°** **44.55**%	5.58cm, 1.48° 43.7%
Kitchen	11.0cm, 2.23° 12.04%	**8.62cm**, 1.95° 22.54%	9.07cm, 1.99° 22.16%	8.67cm, **1.93°** 23.32%	8.73cm, **1.93°** **23.48%**
Stairs	62.9cm, 11.6° 0.2%	**35.6cm**, **6.94°** 0.3%	35.6cm, 7.27° 1.6%	35.9cm, 7.33° 1.5%	36.0cm, 7.25° **1.9%**

Table 5.1: The median pose errors and accuracy for 7Scene dataset of models using different losses. The number ending with *cm* (resp. °) is the median translation (rotation) error for test set. The percentage is the proportion of test frames with both translation and rotation error is below (5*cm*, 5°). The overall performance of the model is significantly improved with the additional constraints provided by the multi-view consistency of the features. Images from scene `stairs` have strong self-similarities, which makes it very difficult to deal with than the rest.

images are used for training, which means the actual scales of the scenes are missing. This heuristic assumes a constant distance between the camera plane and the scene surface for every image. The distance is set to 3m and 10m for 7Scenes and Cambridge Landmarks dataset respectively, which are the approximate scales of the indoor and outdoor scenes.

We apply our proposed multi-view geometry based losses in the second stage of training, which is initialized by the model from the previous heuristic. To conduct a detailed ablation study, we train the model with five different combinations of losses for the 7Scenes respectively:

1. `repro`: the model is trained with only single view reprojection loss equation (5.6). This is our baseline model.
2. `repro+rgb`: the model is trained with photometric reconstruction loss equation (5.1) along with `repro`.
3. `repro+feat`: the model is trained with deep feature reconstruction loss equation (5.4) along with `repro`.
4. `repro+rgb+feat`: the model is trained with photometric reconstruction loss equation (5.1) *and* deep feature reconstruction loss equation (5.4) along with `repro`.
5. `w/ smooth`: the smoothness prior equation (5.5) is added to `repro+rgb+feat`.

All five models are optimized in an end-to-end fashion with ADAM (Kingma et al. 2014) for *30k* iterations in total. The initial learning rate is set to 1e-4 and decreased to half at 10k step and the next every 5k step. The training samples for `repro` model are also pairs of images to ensure an identical training environment. The hyper-parameters in (5.7) are *not* highly tuned and are kept identical between scenes.

We use the PnP solver plus RANSAC to estimate the 6D pose for the test images after predicting the scene coordinate from these 5 trained models. For RANSAC, $N = 256$ pose hypotheses are generated as the pool, and the reprojection

error threshold τ is set to 10 pixels for inliers selection. The final pose refinement step runs up to 100 iterations.

5.4.3 Results Analysis

Table 5.1 and figure 5.3 shows the pose estimation performance of the models trained with different combinations of the losses introduced in previous sections. The pose for a test image is considered as *correct* if the pose error is below 5cm and 5°.

5.4.3.1 Multi-view vs. Single-view

One can see from table 5.1 that the addition of photometric loss supervised by multi-view constraint in training *always* improves the accuracy of the estimated pose than purely training with single-view reprojection loss (Column `repro` and `repro+rgb`). The deep feature reconstruction loss also helps the reprojection loss and the effect is even more obvious generally (Column `repro` and `repro+feat`), due to the more informative (both fine and course) features that are extracted from a deep model, especially when the scene contains textureless regions. The accuracy of the pose estimation is further slightly improved by using the photometric and feature reconstruction loss together as the additional supervision for the coordinate regression model.

The reason behind this gain of pose estimation performance is that the model predicts more accurate scene coordinates if it is supervised with multi-view constraints during training. We show one set of 4 points used by hypothesis generation in PnP algorithm for a test image in figure 5.4. The predicted 3D points for the left image are projected to the right image using the ground truth pose to show the quality of these points. The projections of points from the model with reconstruction loss on the right image are closer to the pixels that share the similar surrounding pattern on the left one, compared to the model trained with only single

Figure 5.4: The projections of scene coordinates predicted by models trained with (*reprojection loss only*) and (*reprojection loss + reconstruction loss*) on a pair of test images. In the left image we show some sample points (coloured circles) for which we predict the 3D coordinates using two models: one trained with single-view reprojection loss and the other trained with the multi-view geometry-based reconstruction loss as the additional supervision. In the right view, whose relative pose to the left is known, we show the projections of the regressed coordinates from left image as squares (reprojection loss) and as stars (geometry loss). Observe that the geometry loss (i.e. with feature consistency constraints), produces a model that produces better coordinates, as seen by the better match locations of the star points compared with the squares. Best viewed in colour.

view reprojection loss. This behavior affirms the usage of photometric/feature reconstruction consistency in the training.

5.4.3.2 Smoothness prior

The best pose estimation performance of scene `heads` comes from the model trained with all components of the final loss, which suggests the best 3D reconstruction. See figure 5.5 for visualization. As can be seen, the point cloud reconstructed from the models trained with the first three (a, b c) losses are not visually good enough to recover the actual geometry of the scene (e). In this case, the smoothness prior (d) helps to produce an improved model for the 3D reconstruction, especially by reducing the noise in the bottom part of the point cloud. As for other scenes, we found the usefulness of the smoothness prior is limited (Column `repro+rgb+feat` and `w/ smooth` in table 5.1). When the pose estimate is accurate enough from the model trained without the smoothness prior, for instance in scenes `fire` and `office`,

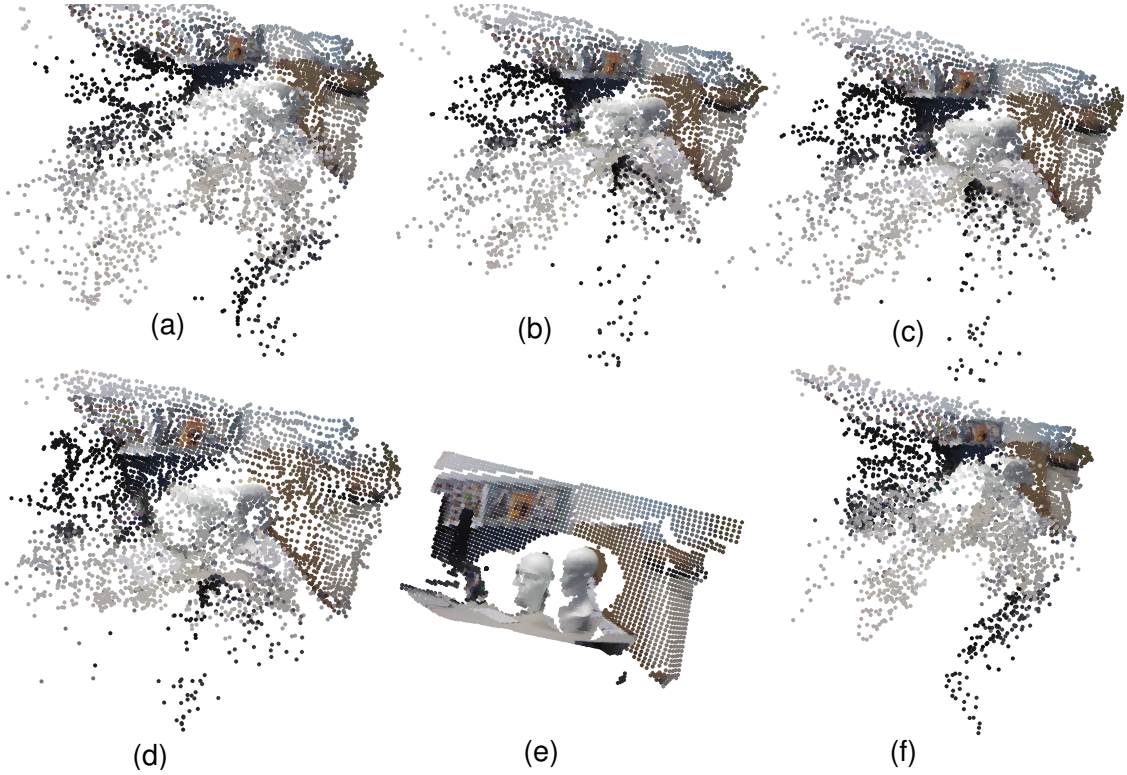


Figure 5.5: Reconstructed point clouds of one sample image from the test set of scene `heads` using different models. We visualize the point cloud reconstruction from our model trained with (**a**) `repro`, (**b**) `repro+rgb`, (**c**) `repro+rgb+feat`, (**d**) `w/ smooth`. The ground truth point cloud and the reconstruction from DSAC++ (Brachmann et al. 2018) is showed in (**e**) and (**f**) respectively. All point clouds are visualized from the same viewpoint.

the use of the smoothness penalty does not help, with the effect even being negative. We presume the underlying reason for this behaviour is that there are relatively more textures in other scenes than `heads`. The losses L_{rgb} and L_{feat} benefit from these textures a lot and therefore recover a good enough scene geometry, even without using smoothness loss. When using penalty of smoothness, the loss might push inlier 3D scene points towards the outliers that would otherwise have been ignored by RANSAC, and subsequently decrease the performance of pose estimation.

5.4.4 Comparison with Single-view Based Work

To establish a fair comparison between our model and other work, we increase the training iteration number of the model `repro+rgb+feat` to 300k (which is used

	DSAC++ Brachmann et al. 2018		ours
Scene	w/ 3D	w/o 3D	
Chess	0.02m, 0.5°	**0.02m**, **0.7°**	**0.02m**, 0.8°
Fire	0.02m, 0.9°	0.03m, 1.1°	**0.02m**, **1.0°**
Heads	0.01m, 0.8°	0.12m, 6.7°	**0.04m**, **2.7°**
Office	0.03m, 0.7°	**0.03m**, **0.8°**	**0.03m, 0.8°**
Pumpkin	0.04m, 1.1°	0.05m, **1.1°**	**0.04m**, **1.1°**
Kitchen	0.04m, 1.1°	0.05m, 1.3°	**0.04m**, **1.1°**
Stairs	0.09m, 2.6°	0.29m, 5.1°	**0.18m**, **3.9°**
Acc.	76.1%	60.4%	**70.1%**
K. Col.	0.18m, 0.3°	0.23m, 0.4°	**0.20m**, **0.3°**
Old Hos.	0.20m, 0.3°	0.24m, 0.5°	**0.19m**, **0.4°**
Shop Fac.	0.06m, 0.3°	0.09m, 0.4°	**0.07m, 0.3°**
St M. Ch.	0.13m, 0.4°	**0.20m**, 0.7°	**0.20m**, **0.6°**
G. Court	0.40m, 0.2°	0.66m, **0.4°**	**0.62m**, **0.4°**

Table 5.2: Comparison between our method and DSAC++ (Brachmann et al. 2018). The gap between model trained with and without is closed using our multi-view geometry-based training method. Numbers are boldened only among the w/o 3D methods.

in Brachmann et al. 2018), likewise for the steps for learning rate decay. Table 5.2 shows the results of this model for 7Scenes and Cambridge Landmarks (Kendall et al. 2015). Except for the relatively poor performance in the `stairs` scene due to the self-similarity of the RGB images, our method achieves a consistently good result for all of the indoor scenes. The percentage of the correct test frames of all scenes in 7Scenes of our model is 70.1%, compared to 76.1% and 60.4% of DSAC++ (Brachmann et al. 2018)'s model that is trained with and without ground truth scene coordinate respectively. The gap of training without and with the 3D model of the scene is closed by our method.

The conclusions from previous ablation study in table 5.1 and table 5.2 together are: 1) our losses help the coordinate regression model converge faster than the single-view baseline (table 5.1). This superiority of convergency is indicated by the better accuracy of the model trained with our proposed multi-view loss than the single-view baseline, when both models are optimized for 30k iterations. 2) when converged (300k iterations in table 5.2), it also performs better than the

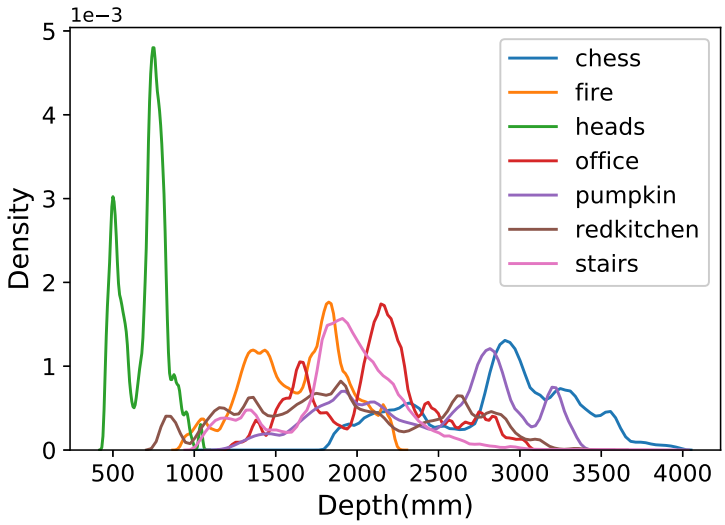


Figure 5.6: The distribution of depth value of 7Scenes. We randomly select 10 depth images from the training set of each scene, and show the distributions of all the valid depth values of them. One can see that the depth distribution of scene `heads` has the mean value around 0.7m, which does not follow the distributions of other scenes.

state-of-the-art single-view method (Brachmann et al. 2018) (table 5.2).

A noticeable point is that the median error of scene `heads` is relatively large for DSAC++ (Brachmann et al. 2018) compared to other indoor scenes when trained without 3D model, which is 12cm and 6.7°. We observe that this is presumably due to the misused heuristic for scene `heads`, which assumes a constant distance between the image plane and the scene surface for every frame that is used to initialize the model in the first stage of training. To support our hypothesis, we plot the distributions of the ground truth depth samples from training images of each scene in figure 5.6. The heuristic constant distance we (as well as DSAC++ (Brachmann et al. 2018)) used for 7Scenes is 3m, which properly simulate the substantial depth of most of the scenes, except for `heads`, whose true depth is around 0.7m. We therefore train another model for `heads` with the heuristic set to 0.7m. The result of the new model is increased to 0.02m and 1.3°. This backs our speculation. Nonetheless, our training scheme eliminates the negative effect of the inappropriate heuristic, and achieves better reconstruction when the poor prior is applied to both our method and Brachmann et al. 2018 (we still use 3m for 7Scenes as the approximate depth

*For all test images in 7Scenes	DSAC++Brachmann et al. 2018	Ours
Average No. of inliers per image	245	**319**

Table 5.3: We project the predicted scene coordinates from the models in DSAC++Brachmann et al. 2018 and ours using ground truth poses. The reprojection error threshold for inlier is set to 2 pixel.

for the first stage in our experiment). From this standpoint, our method based on multi-view consistency reduces the dependence on the initialization of the model.

Since the performance of pose estimation heavily relies on the quality of the scene coordinate prediction, we also show the quantitative comparison of the scene coordinate regressed by our model trained with multi-view constrains (`repro+rgb+feat`) and the single-view method (Brachmann et al. 2018) in table 5.3. This shows that our model predicts more accurate scene coordinates for the geometrical task.

5.5 Conclusion

We have proposed an efficient learning method for scene coordinate regression to carry out accurate 6DoF camera relocalisation in a known scene from a single RGB image. Our learning method explicitly enforces multi-view geometric constraints to learn the regression model in a self-supervised manner in the absence of the ground truth 3D model. The constraints imposed by our proposed loss improve the efficiency of training. Additionally, the regression model learned via our method allows for more reliable 2D-3D correspondences which in turn lead to consistent and accurate camera relocalisation performance.

In next chapter, we aim to apply the proposed multi-view constraints to object pose estimation task with specific modifications.

Bibliography

Sattler, Torsten, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe (2019). "Understanding the limitations of cnn-based absolute camera pose regression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3302–3312.

Shotton, Jamie, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon (2013). "Scene coordinate regression forests for camera relocalization in RGB-D images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937.

Brachmann, Eric, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother (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*, pp. 3364–3372.

Brachmann, Eric, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother (2017). "Dsac-differentiable ransac for camera localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6684–6692.

Brachmann, Eric and Carsten Rother (2018). "Learning less is more-6d camera localization via 3d surface regression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4654–4662.

Bui, Mai, Shadi Albarqouni, Slobodan Ilic, and Nassir Navab (2018). "Scene coordinate and correspondence learning for image-based localization". In: *arXiv preprint arXiv:1805.08443*.

Gao, Xiao-Shan, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng (2003). "Complete solution classification for the perspective-three-point problem". In: *IEEE transactions on pattern analysis and machine intelligence* 25.8, pp. 930–943.

Lepetit, Vincent, Francesc Moreno-Noguer, and Pascal Fua (2009). "Epnp: An accurate o (n) solution to the pnp problem". In: *International journal of computer vision* 81.2, p. 155.

Hesch, Joel A and Stergios I Roumeliotis (2011). "A direct least-squares (DLS) method for PnP". In: *2011 International Conference on Computer Vision*. IEEE, pp. 383–390.

Wang, Ping, Guili Xu, Zhengsheng Wang, and Yuehua Cheng (2018). "An efficient solution to the perspective-three-point pose problem". In: *Computer Vision and Image Understanding* 166, pp. 81–87.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Sattler, Torsten, Bastian Leibe, and Leif Kobbelt (2016). "Efficient & effective prioritized matching for large-scale image-based localization". In: *IEEE transactions on pattern analysis and machine intelligence* 39.9, pp. 1744–1756.

Hartley, Richard and Andrew Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Weerasekera, Chamara Saroj, Ravi Garg, and Ian Reid (2017). "Learning deeply supervised visual descriptors for dense monocular reconstruction". In: *arXiv preprint arXiv:1711.05919*.

Zhan, Huangying, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid (2018). "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 340–349.

Garg, Ravi, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid (2016). "Unsupervised cnn for single view depth estimation: Geometry to the rescue". In: *European Conference on Computer Vision*. Springer, pp. 740–756.

Zhou, Tinghui, Matthew Brown, Noah Snavely, and David G Lowe (2017). "Unsupervised learning of depth and ego-motion from video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858.

Jaderberg, Max, Karen Simonyan, Andrew Zisserman, et al. (2015). "Spatial transformer networks". In: *Advances in neural information processing systems*, pp. 2017–2025.

Silberman, Nathan, Derek Hoiem, Pushmeet Kohli, and Rob Fergus (2012). "Indoor segmentation and support inference from rgbd images". In: *European conference on computer vision*. Springer, pp. 746–760.

Godard, Clément, Oisin Mac Aodha, and Gabriel J Brostow (2017). "Unsupervised monocular depth estimation with left-right consistency". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279.

Heise, Philipp, Sebastian Klose, Brian Jensen, and Alois Knoll (2013). "Pm-huber: Patchmatch with huber regularization for stereo matching". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2360–2367.

Kendall, Alex, Matthew Grimes, and Roberto Cipolla (2015). "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946.

Ilg, Eddy, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox (2017). "Flownet 2.0: Evolution of optical flow estimation with deep networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2462–2470.

Reda, Fitsum, Robert Pottorff, Jon Barker, and Bryan Catanzaro (2017). *flownet2-pytorch: Pytorch implementation of FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks*. https://github.com/NVIDIA/flownet2-pytorch.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

6

Reconstruct Locally, Localize Globally: A Model Free Method for Object Pose Estimation

Contents

In this chapter we return to the question of 6dof pose estimation for objects. Although this problem is classically posed as a correspondence problem between a known geometric model, such as a CAD model, and image locations, in this chapter we consider the case that instead of a CAD model, we have access to some previously

acquired images of the object. Instead of creating an intermediate 3D object model by reconstructing from the previous views (as in (Wagner et al. 2008; Pan et al. 2010; Pan et al. 2009)), we propose a learning-based method whose input is the collection of previous images of the target object, and whose output is the pose of the object in a novel view. At inference time, our method maps from the RoI features of the input image to a dense collection of object-centric 3D coordinates, one per pixel. This dense 2D-3D mapping is then used to determine 6dof pose using standard PnP plus RANSAC. In this sense the work in this chapter can be considered an extension of the work of chapter 5 to the problem of chapter 3. It differs from chapter 5 in that we also introduce a mechanism to automatically discover and match image features that are consistent across the multiple prior views. We show that this method eliminates the requirement for a 3D CAD model (needed by classical geometry-based methods and state-of-the-art learning based methods alike) but still achieves performance on a par with the prior art. This work was recently accepted by the IEEE Conference on Computer Vision and Pattern Recognition 2020.

6.1 Introduction

In chapter 3, we proposed a method for the 6dof object pose estimation based on direct regression. It performs fast, is reasonably accurate and does not require any geometric information about the objects. However the classical feature-based solutions (Dementhon et al. 1995; Zhang 1994; Marchand et al. 1999; Lepetit et al. 2005; Pauwels et al. 2013) and recent learning-based publications (Xiang et al. 2017; Li et al. 2018; Sundermeyer et al. 2018; Hu et al. 2019; Peng et al. 2019; Wang et al. 2019; Xiao et al. 2019; Bui et al. 2018; Park et al. 2019) in this field suggests that if the 3D CAD models of objects are known, it would help algorithms to achieve better accuracy.

3D CAD model has been widely used for object pose estimation and played different roles. For instance, in the classic family, such a role might be the reference for registration (Zhang 1994), base for templates generation (Hinterstoisser et al. 2012) and provider of texture for feature extraction. As for the CNN-based approaches, this model acts as the supervision for network learning (Brachmann et al. 2014; Park et al. 2019; Brachmann et al. 2016; Kehl et al. 2017), a source for synthetic image generation (Peng et al. 2019; Li et al. 2018; Kehl et al. 2017; Chang et al. 2015) and/or an agent for post-process refinement (Li et al. 2018; Rad et al. 2017; Kehl et al. 2017) *etc.* However, fine-grained and well-textured 3D structure does not exist for every object in the wild. This limits the generalization of these approaches. In this chapter, we are therefore devoted to answer this question: Is it possible to accomplish the object pose estimation task based on strong geometry learning, *but without using the 3D CAD model of the object?*

Methods based on object reconstruction (Wagner et al. 2008; Pan et al. 2010; Pan et al. 2009) have shown the feasibility of this proposal. They firstly reconstruct the 3D object from the multi-view RGB images to substitute missing CAD model, using Structure from Motion (SfM) (Agarwal et al. 2011). Object pose is then solved using the Perspective-n-Point (PnP) algorithm, based on the correspondence of the 2D visual cues of a new image and those affiliated to the 3D reconstruction. Although handcrafted feature descriptors perform efficiently in detection and matching, they cause the main limitations in the pipeline: (i) Their main purpose is to generally detect the salient keypoints with rich texture, rather than to describe the structure of the object; (ii) For largely texture-less objects, a paucity of interest points can often lead to a poor or unreliable interpolated reconstruction.

Camera relocalisation, as we have discussed in previous chapters, is a very closely related problem (because its objective is also to find a 6dof pose), and this has been tackled using regression-based method like PoseNet (Kendall et al. 2015). More promising are the methods of Brachmann et al. 2017; Brachmann

et al. 2018 and chapter 5 which use the power of CNNs to establish high quality dense correspondences and are coupled with a subsequent geometric method for improved accuracy. Nevertheless there are aspects of the camera relocalisation problem that are not directly analogous to object pose estimation. The main difference that prevents direct adoption of these methods for object pose is that the object is only visible in part of the scene, necessitating a need to distinguish the object from the rest of the scene.

Hence, the problem we seek to solve is: given as input a collection of images and their poses, learn a system that can then detect, reconstruct and localize the object in any subsequent view. Inspired by the success of the hybrid approach (Brachmann et al. 2014; Brahmbhatt et al. 2018; Bui et al. 2018), we introduce: Reconstruct Locally, Localise Globally (RLLG), a learning and reconstruction-based method to object pose estimation.

Our solution differs from SfM in that there is no explicit 3D model of the target created. We implicitly encode the process of reconstruction within the weights of a neural network during training. At inference stage, this network serves as a 2D-3D correspondences establisher for the test image. Our method then estimates the accurate 6dof pose of the object from the these correspondences using PnP plus RANSAC (Fischler et al. 1981).

In order to identify, detect and isolate the objects from the background, and concurrently perform reconstruction, we (again, like chapter 3) seamlessly build our model upon Mask R-CNN (He et al. 2017). Along with three special-purpose heads: bounding box head, classification head and segmentation head, we contribute a new head – the *object coordinate head* – to the same backbone, whose output is the dense 3D coordinates of the object in object-centric frame. In this sense, it is the local analog of what we saw in chapter 5, every pixel within the foreground is regressed to its 3D coordinate by the object coordinate head.

But, a key issue here is how to provide supervision to learn this head. Since the goal of RLLG is to disengage the ground truth 3D model from the pose estimation pipeline, how to establish what those 3D coordinates actually are is our main interest. Similar to the previous chapter, we continue to explore alternative supervisions from multi-view geometry. In chapter 5, the geometric supervision of the 3D coordinates came directly from a photometric (and featuremetric) warp loss. However, because the objects are small and sometimes textureless, in this chapter, we will augment the photometric loss with a warp loss based on the equivariant features, *i.e.* features that are not affected by a change in viewpoint. To that end, we are going to introduce an additional component to the learning framework, which explicitly transforms each local image patch into an equivariant feature. We will learn it so that at training time, for the object coordinate learning, dense 2D-2D correspondences can be established which enable the formulation of multi-view geometry to provide the training signal. The equivariant feature branch is not to be used at inference time for a single-view image, since the network already knows how to regress to 3D object coordinates from pixels after training.

In summary, we design the object coordinate head as a two-branched FCN (Long et al. 2015) during training. The equivariant feature branch learns dense viewpoint-independent features for all pixels on the object, and those features are matched between pairs of images in the training set to build 2D-2D correspondences. The object coordinate branch regresses to the 3D coordinates in the object-centric frame for every object pixel. This regression is learned using a loss based on the multi-view geometry according to the matching results.

As mentioned before, one of the key issues we would like to solve in this chapter is to learn object coordinates without using direct supervision. This issue also applies to the equivariant feature branch. Since these feature descriptors are also invariant to the change of the external factors (such as pose and illumination), the learning therefore aligns them in pairs of images related by a warp and expects the

detector to be equivariant with the image deformations. In (Thewlis et al. 2019), the authors use in-plan transformations (e.g. in-plane rotation, scaling and crop) for data augmentation as a way to learn equivariant features. We follow the same idea and use these artificial deformations to build *equivariant constraints*.

A question that the reader may raise is why do we not artificially create an image-pair with known deformation (same as what is used for equivariant feature learning) to enforce multi-view geometry for object coordinate learning, but use a 'redundant' equivariant feature branch to find the 2D-2D correspondences explicitly for images from actually different viewpoints in the training set. The reason is that from a geometric perspective, the pixel-wise correspondences between an image-pair introduced by the in-plane operations do *not* constrain the location of the object point in 3D space. Therefore, we propose to learn the object coordinate with image pairs derived by out-of-plane movements, and used the equivariant feature branch to assist the establishment of correspondences between images.

We have created a dataset to showcase our object coordinate regression network and subsequent pose estimation pipeline. Our 3D model free pose estimation method is also tested on the LINEMOD (Hinterstoisser et al. 2012) and Occlusion LINEMOD (Hinterstoisser et al. 2012) dataset to prove its generalization and robustness to real world scenarios. It achieves the on-par performance with the state-of-the-art methods that require the 3D object in different ways.

6.2 Related work

In conjunction with section 3.2 of chapter 3, we review the learning-based methods that rely on the 3D CAD model of object to perform pose estimation in this section.

Like detection, segmentation and other recognition tasks, object pose estimation also benefits from the recent development of deep learning. Most of the learning-based methods integrate the 3D object model in the process of learning and/or

inference. BB8 (Rad et al. 2017), Oberweger et al. 2018, Tekin et al. 2018 and Hu et al. 2019 create a 3D bounding box around the object model, and define the 8 (or 9 with center point in (Tekin et al. 2018)) corners as the 3D key-points on the object. They then annotate their 2D projections and train various networks to perform keypoint detection on an image, establishing sparse 2D-3D correspondences for pose estimation. PVNet (Peng et al. 2019) proposes a method that automatically discovers a set of keypoints on the 3D object surface based on the physical structure, to ensure that their 2D projection are all within the silhouette.

The CAD model is also very handy when generating new data for the training. Peng et al. 2019; Rad et al. 2018; Oberweger et al. 2018 use the textured object model and random poses to generate a large amount of synthetic images to augment (or replace) the limited training images, preventing the network from overfitting. The 3D object model could also serve as the base for loss evaluation. Wang et al. 2019; Li et al. 2018; Xiang et al. 2017 compares the offsets between the object model transformed by the predicted pose and the ground truth pose. This error is used for back-propagation to train the network, and successfully avoids the imbalanced weighting between translation and rotation when a model builds the losses using distances in the translational and rotational spaces separately (such as Kendall et al. 2015 and Kendall et al. 2016).

Moreover, in (Xiao et al. 2019; Li et al. 2018; Rad et al. 2017; Kehl et al. 2017), the 3D model is used for post-refinement to improve the quality of the pose estimates. Having the output pose from the network as the initialization, an iterative optimization is designed to produce the optimal pose solution by minimizing an objective related to the 3D model. Such objective can be the consistency between the rendered colour image from the textured model and the input image (Rad et al. 2017), or the distance between the transformed object points in camera frame and those recovered from depth (Kehl et al. 2017).

Similar to our work, Pix2Pose (Park et al. 2019) and Brachmann et al. 2014 also use object coordinates as an intermediate representation to find the object pose. However, in these methods, a 3D model of the object provides the direct supervision for the model (such as a random forest (Brachmann et al. 2014) or a neural network (Park et al. 2019)) learning. In contrast, we aim to learn the coordinates without the 3D model in a self-supervised way (by self-supervised, we mean that the supervision that governs the learning of the object coordinate does not come from the ground truth directly).

6.3 Method of Reconstruct locally, Localize globally

Denote by $I_i, i \in \{1...n\}$ an image of object O_l, where $l \in \{1...L\}$ is object label, and by $\mathbf{P}_{i,l}$ the visible 3D object points in I_i. Their coordinates in object-centric frame O and camera-centric frame C are $\mathbf{P}^O_{i,l}$ and $\mathbf{P}^C_{i,l}$ respectively. The pose of this object $\mathbf{T}_{i,l}$ consists of two parts: rotation $\mathbf{R}_{i,l} \in \mathbb{R}^{3\times3}$ and translation vector $\mathbf{t}_{i,l} \in \mathbb{R}^3$. It is essentially the transformation between two Euclidean spaces:

$$\mathbf{P}^C_{i,l} = \mathbf{R}_{i,l}\mathbf{P}^O_{i,l} + \mathbf{t}_{i,l}. \tag{6.1}$$

Camera intrinsics $\mathbf{K}$ projects $\mathbf{P}^C_{i,l}$ onto image and obtains the 2D coordinates of the projections $\mathbf{p}_{i,l} = [\mathbf{u}, \mathbf{v}]$, where

$$s\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\\ 1\end{bmatrix} = \mathbf{K}\mathbf{P}^C_{i,l} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix}f_x & 0 & c_x\\ 0 & f_y & c_y\\ 0 & 0 & 1\end{bmatrix}. \tag{6.2}$$

s is a scale factor, f_x and f_y are the focal lengths and (c_x, c_y) is the camera center.

The correspondences between 2D points $\mathbf{p}_{i,l} = [\mathbf{u}, \mathbf{v}]$ and 3D points $\mathbf{P}^O_{i,l}$ preserve the geometric transformation of the object to the camera, and therefore are used to estimate the pose at inference time. We aim to design a network to densely build

these correspondences, by mapping from the RGB image pixels to 3D coordinates in the object space.

6.3.1 Object Coordinate Head

We build our object coordinate head upon Mask R-CNN (He et al. 2017). For details of this framework in our thesis, please refer to section 3.3.1.

Figure 6.1 shows the training of the proposed object coordinate head. As mentioned in section 6.1, this new head consists of two branches: object coordinate branch and equivariant feature branch. The object coordinate branch is introduced first since it is directly related to the subsequent pose estimation at the inference time. We then discuss the equivariant feature brach and show how it is learned, and used at training time to benefit the learning of the object coordinates without a strong supervisory signal.

6.3.1.1 Object Coordinate Branch

The spatial map of the object coordinate relates to the 2D layout of the object in the image. Therefore by nature we use convolutions to provide the pixel-to-pixel correspondences between image and object coordinates. We apply a FCN Φ_{obj} on each RoI features. The output of Φ_{obj} is a $m \times m \times 3$ vector map $\mathbf{P}^{O}_{i,l,(h,w)} = \Phi_{obj}(I_i)$, $h \in \{1 \dots m\}, w \in \{1 \dots m\}$, where each pixel is a 3D vector that represents a location on the imaginary 3D model of the target object.

The training of Φ_{obj} is straightforward if the 3D object model is accessible, which makes the learning fully supervised. Instead, we aim to present a model-free method and therefore propose to explore alternative supervisions.

6.3.1.2 Single-view Reprojection Loss

Due to the graceful alignment provided by the FCN, the predicted object coordinate map maintains the explicit per-pixel spatial correspondence with the RoIs, which essentially means that for a specific pixel, the regressed 3D coordinate from the

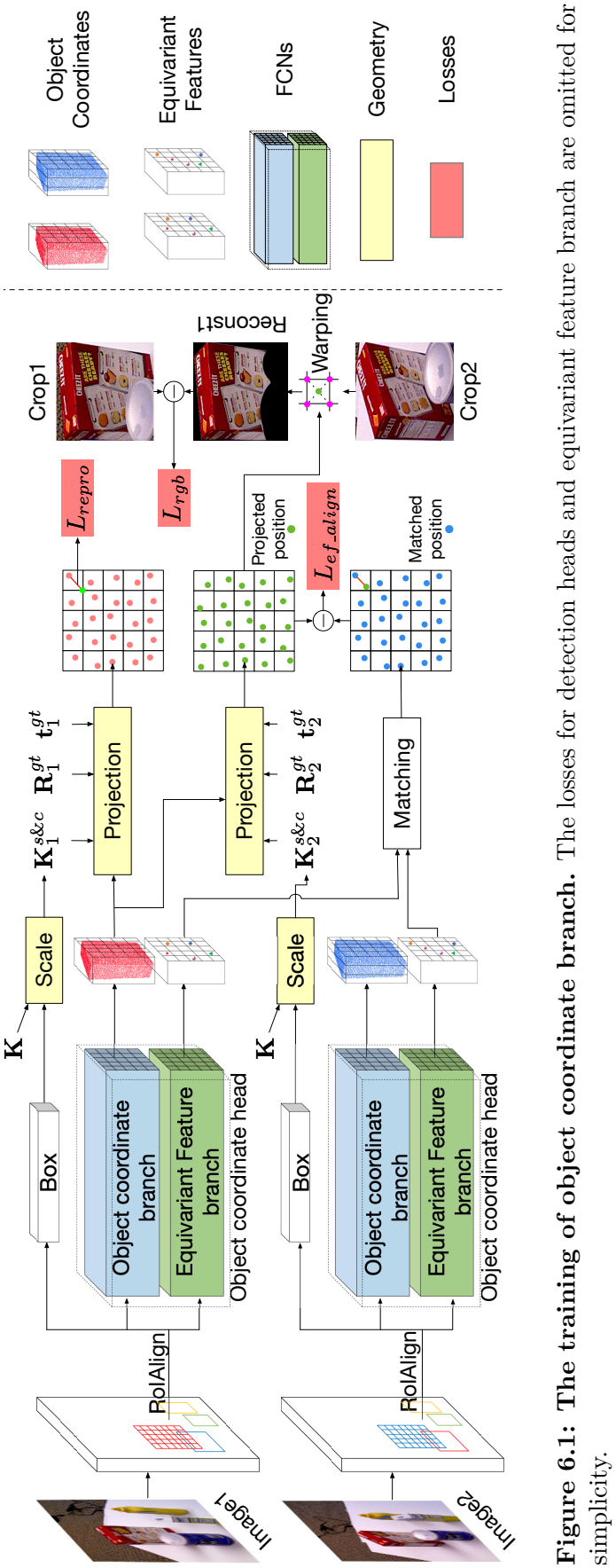


Figure 6.1: The training of object coordinate branch. The losses for detection heads and equivariant feature branch are omitted for simplicity.

network should be re-projected to its exact location using the ground truth pose. We first explore the supervision based on this reprojection error, within the proposal boxes (RoIs) suggested by RPN of Mask R-CNN.

To perform projection inside the RoIs, we need to adapt the projection matrix $\mathbf{K}$ to a proposal box. For each proposal, the RPN estimates a 4D vector $(x_{min}, y_{min}, x_{max}, y_{max})$ that parameterizes a box around the target pixel. In term of spatial dimension, with this box, the RoIAlign layer gathers and pools the RoI features from the backbone and then up/down-samples to $m \times m$ via the FCN Φ_{obj}. Two operations change the spatial dimension of our interested region and consequently reform the projection model: crop (by the RoIAlign) and resize (by up/down-sampling). We therefore assume the $m \times m$ object point map fully corresponds to a *new* $m \times m$ image $I_{i,s\&c}$, which is a resized crop of I_i. The intrinsics hence scales to

$$\mathbf{K}_{c\&s} = \begin{bmatrix} s_w f_x & 0 & s_w(c_x - x_{min}) \\ 0 & s_h f_y & s_h(c_y - y_{min}) \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.3}$$

where $s_w = m/(x_{max} - x_{min})$ and $s_h = m/(y_{max} - y_{min})$. As a result, the predicted re-projection on $I_{i,s\&c}$ from ground truth object pose is

$$\mathbf{p}^{pred}_{i,l,(h,w)} = \frac{1}{s}\mathbf{K}_{c\&s}(\mathbf{R}^{gt}_{i,l}\mathbf{P}^{O}_{i,l,(h,w)} + \mathbf{t}^{gt}_{i,l}). \tag{6.4}$$

The expected projection of an object coordinate simply is the 2D pixel position where it lies in the output map, which means $\mathbf{p}^{gt}_{i,l,(h,w)} = [h, w], h \in \{1 \dots m\}, w \in \{1 \dots m\}$. The learning objective is to minimize the reprojection error triggered by any difference that we assume arises from an error in the predicted object coordinates. We therefore define the single-view reprojection loss as

$$L_{repro} = \frac{1}{m \times m} \sum_{h,w} \left\| \mathbf{p}^{pred}_{i,l,(h,w)} - \mathbf{p}^{gt}_{i,l,(h,w)} \right\|_2 . \tag{6.5}$$

Since loss (6.5) is evaluated for a single-view image, it potentially has limitations. From a geometric perspective, the reprojection loss settles to be optimal for any

point on the line that connects the camera origin and the real 3D object point. Hence, theoretically, minimizing loss (6.5) dose not guarantee the network to regress to the correct coordinates. The training however happens iteratively in practice, which means that the network sees images of the object in different viewpoints from batch to batch. It is expected that the network learns to recognize the same object point with various visual appearance (caused by viewpoint change) in different images and *consistently* regress to a same coordinate. Such behavior would be an implicit multi-view constraint for the learning and contributes to discover the true geometry of the object.

In order to experimentally validate this hypothesis, we create a synthetic dataset (the details are given in section 6.4) and train the object coordinate head with loss term (6.5). The trained model is tested with an object image and its rotated variant (figure 6.2(a)). The predicted object points are shown in figure 6.2(b) in red and blue respectively. The obvious incompatibility in two reconstructions, which is indicated by the point-wise differences between points in the red and blue point cloud, suggests that single-view loss-based training does not produce a consistent 3D coordinate for the same object point in different views.

6.3.1.3 Mult-view Geometry-based Loss

To the limitation addressed in the previous section, we propose to make the multi-view constraints explicit and provide strong geometric supervision for object coordinates learning. Based on Hartley et al. 2003, images from multiple viewpoints can be used to constrain the coordinate for a 3D point using triangulation. Such geometry is built upon the 2D-2D correspondences between the objects in different images. To that end, we propose to include an additional *equivariant feature branch*, which discovers equivariant feature for an image patch of the object. The learned features for multiple images are then matched during the learning of object

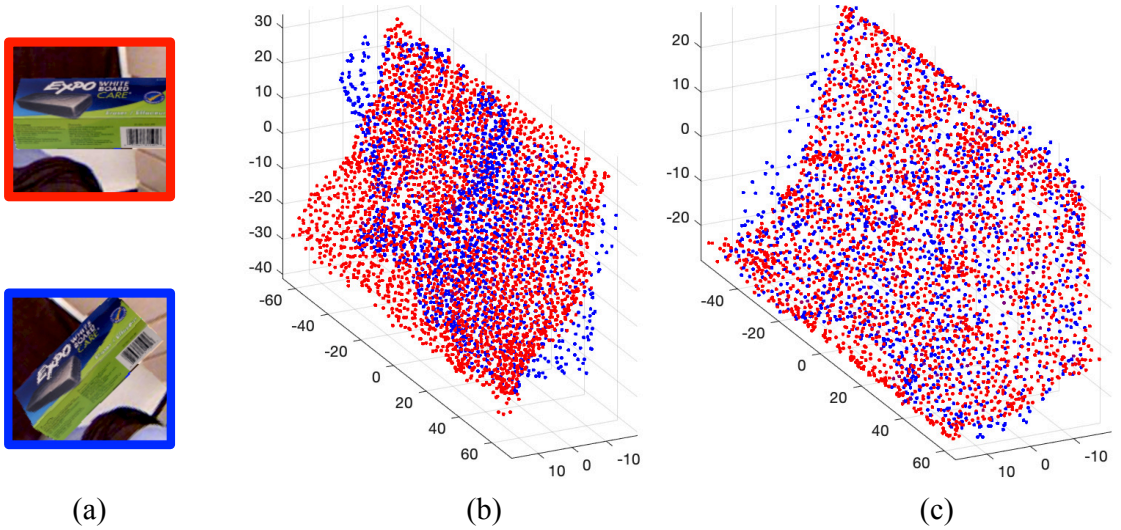


Figure 6.2: Comparison between the 3D object points for an image and its variant. (a): an object image and its rotated version. (b): Reconstruction from single-view reprojection loss. (c): Reconstruction from multi-view consistence loss.

coordinates, establishing a dense collection of 2D-2D correspondences. The multi-view constraints are explicitly built accordingly.

Equivariant Feature Branch. Equivariant feature is defined as the feature vector for a patch of the object image that can be recognized and correlated from different viewpoints. Its representation is a d dimensional feature vector learned automatically by the network for uniqueness and rich descriptiveness. It is intrinsic to the object, which means the change of viewpoint or deformation should not cause any difference to the representation of a unique feature on the object. Such behavior is defined as *equivariance constraint* (Thewlis et al. 2017). We therefore exploit this property as the supervision for equivariant feature learning due to the lack of manual annotation.

The equivariant feature branch is also a FCN because of the one-to-one mapping from image pixels to equivariant features. It is learned in a Siamese configuration with two images – I_i and $r(I_i)$ – correlated by a known deformation r. Such deformation transforms the point (h, w) of the source to (h_r, w_r) on the target. (See figure 6.3 for an example of the feature for a single image point.)

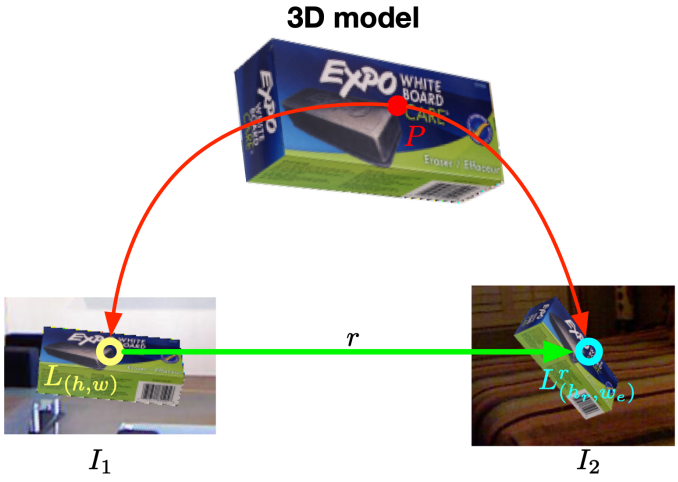


Figure 6.3: Equivariance constraint for object feature. As an example, P is one of the points on the object surface. It is projected to two images. These two images are corelated by a known deformation r. The equivariant features of these projections, $L_{(h,w)}$ and $L^r_{(h_r,w_r)}$), must be compatible with this deformation. Please note that the usage of 3D model in this figure only serves the purpose of illustrating the characteristic of equivariant feature, we do *not* use it in our pose estimation pipeline.

Denote by Φ_{ef} the equivariant feature branch. It takes I_i and $r(I_i)$ as input at the same time and outputs two $m \times m \times d$ feature maps $L = \Phi_{ef}(I_i)$ and $L^r = \Phi_{ef}(r(I_i))$ for each RoI. The equivariance constraint is defined as $L_{(h,w)} = L^r_{(h_r,w_r)}$ where $h, w \in 1 \dots m$. In order to prevent this constraint from falling into a degenerated case, when all the pixels are mapped to a singular object feature, we follow Thewlis et al. 2019 to reformulate it to a distance-aware softmax loss.

The relative similarities between the equivariant features on two RoIs are formulated by a softmax function on the cosine similarities. What is expected from the learning is that the equivariant features on two images with short spatial distance have large similarity, and vice versa. To be more specific, for example, the features of pixels that are close to (h_r, w_r) on image $r(I_i)$ should be similar to the feature $L_{(h,w)}$, which is generated for pixel (h, w) on I_i, whereas the features of pixels that are far away from (h_r, w_r) should be dissimilar to $L_{(h,w)}$. Therefore the relative similarities

of features in two views are weighted by the spatial distances in the loss term

$$L_{ef} = \frac{1}{m^4} \sum_{\substack{h_s,w_s \\ h_t,w_t}}^{m} \text{dist}(s,t) \frac{e^{s((h_s,w_s),(h_t,w_t))}}{\sum\limits_{h'_t,w'_t}^{m} e^{s((h_s,w_s),(h'_t,w'_t))}}, \tag{6.6}$$

where $\text{dist}(s,t) = \|(h_s,w_s) - (h_t,w_t)\|_2$, and

$$s((h_s,w_s),(h_t,w_t)) = \frac{L_{(h_s,w_s)} \cdot L^r_{(h_t,w_t)}}{\left\|L_{(h_s,w_s)}\right\|_2 \left\|L^r_{(h_t,w_t)}\right\|_2} \tag{6.7}$$

is the cosine similarity between the equivariant features.

There are various of choices for deformation r to benefit the learning of the equivariant features. Nonetheless we consider the in-plane rotation and scaling (to ensure a same dimension with the original image), which preserve the rigidness of the object.

Thanks to the uniqueness of the learned equivariant feature descriptors, they can be matched from two images that are related by an out-of-plane movement. The following paragraphs show our method of incorporating the matched 2D-2D correspondences into a multi-view loss term.

Multi-View Loss. With the motivation of introducing multi-view geometry into learning, we upgrade the object coordinate branch to a Siamese configuration as well. We use two images I_s and target I_t – from different viewpoints caused by an out-of-plane movement in the training set – as the inputs for the Siamese network.

The proposed multi-view loss for the object coordinate branch consists of two terms. First, we focus on the cross-projection between two viewpoints. Given I_s and I_t as the inputs for the object coordinate branch *and* equivariant feature branch, four outputs are obtained: object coordinate maps $\Phi_{obj}(I_s), \Phi_{obj}(I_t)$ and feature maps $\Phi_{ef}(I_s), \Phi_{ef}(I_t)$. Pixel-wise matching is performed on these feature maps. Denote by $\mathbf{p}^{ef}_{t,l,(h,w)} = M(\Phi_{ef}(I_s), \Phi_{ef}(I_t))$ the matched position of I_s's pixel on I_t, where the M is a matching operation. Given the ground truth pose of the

target image $\mathbf{R}_t^{gt}, \mathbf{t}_t^{gt}$ and the scaled camera matrix $\mathbf{K}_t$, the projection of predicted source object points on the target RoI is

$$\mathbf{p}_{t,l,(h,w)}^{proj} = \frac{1}{s}\mathbf{K}_t(\mathbf{R}_t\mathbf{P}_{s,l,(h,w)}^{O} + \mathbf{t}_t). \tag{6.8}$$

$\mathbf{p}_{t,l,(h,w)}^{proj}$ and $\mathbf{p}_{t,l,(h,w)}^{ef}$ are the position of a same 3D object point on the target RoI. The difference between them is used for back-propagation to learn a 3D coordinate whose projection agrees with the matched position. Thus the first loss term is defined as the equivariant feature alignment loss:

$$L_{ef\_align} = \frac{1}{m \times m}\sum_{h,w}\left\|\mathbf{p}_{t,l,(h,w)}^{proj} - \mathbf{p}_{t,l,(h,w)}^{ef}\right\|_2. \tag{6.9}$$

Please refer to L_{ef_align} in figure 6.1 to see the illustration of feature alignment loss.

Second, we propose to encode the multi-view constraints as a photometric loss. Specifically, the projections $\mathbf{p}_{t,l,(h,w)}^{proj}$ warp a reconstructed image $I_{s\leftarrow t}$ from I_t. Any difference that we assume arises from an error in the predicted object coordinates leads to an error in the normalized RGB space. This behavior encodes a photometric loss:

$$L_{rgb} = \frac{1}{m \times m}\sum_{h,w}\left\|I_{s\leftarrow t} - I_s\right\|. \tag{6.10}$$

Please refer to L_{rgb} in figure 6.1 to see the illustration of photometric loss.

Our multi-view geometry-based loss ultimately is

$$L_{multi} = L_{ef\_align} + L_{rgb}. \tag{6.11}$$

These strong geometric supervisions improve the consistence for the object coordinate regression. Reconstructed results in figure 6.2(c) show the improvement, in which two sets of object points are well aligned for images from different views.

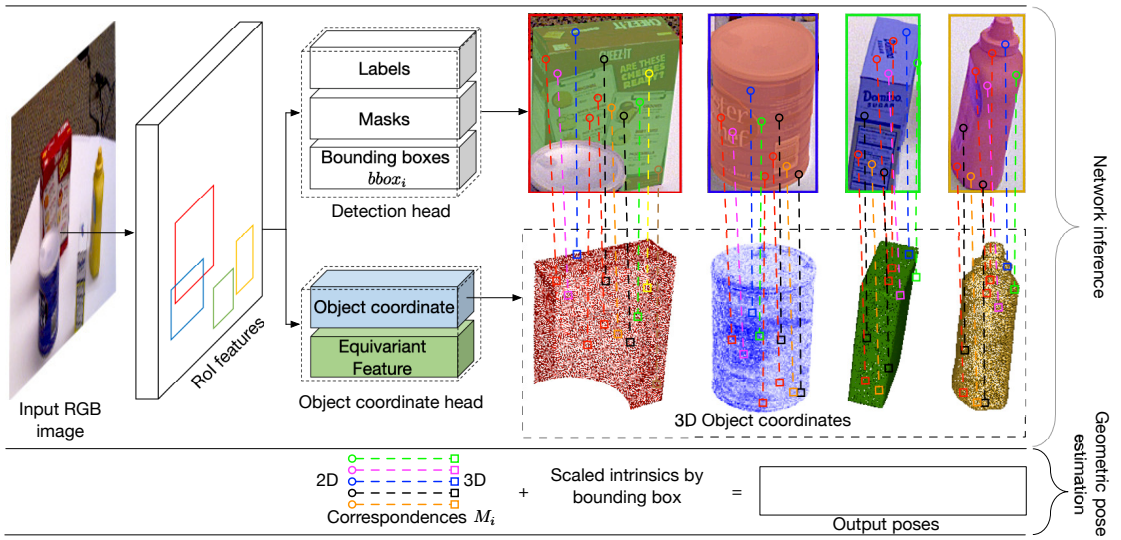


Figure 6.4: The inference of our approach.

6.3.2 Inference

We show the process for pose inference in figure 6.4. Although our model is trained with pairs of images, it requires only a single image to infer the object pose. For a novel image, the detection head predicts a box and a mask for the object. Meanwhile the object coordinate head outputs a 3D object point map for the box. The object points on the background are removed according to the mask. The remaining points are used for establishing the 2D-3D correspondences within the box. The 6dof pose is then solved via PnP plus RANSAC based on these correspondences along with the scaled projection matrix derived from the box position. The pose estimate is subsequently refined using the predicted object points. The equivariant feature branch is turned off at inference time.

6.3.3 Implementation Details

The backbone for RPN in our implementation is ResNet-50 with Feature Pyramid Network (FPN) (Lin et al. 2017). See the details of the detection and segmentation head in He et al. 2017. The architecture of our object coordinate branch is shown in figure 6.5. We follow the structure of the *SmallNet* in Thewlis et al. 2017; Thewlis

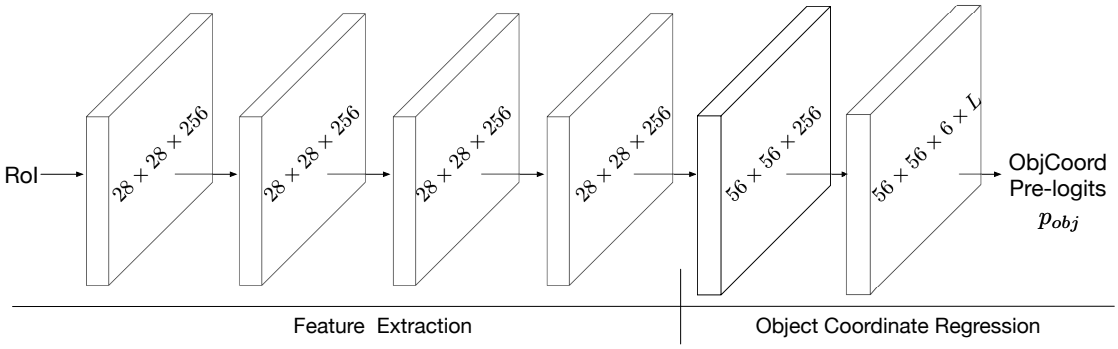


Figure 6.5: Object Coordinate Head Architecture. The feature extractor comprises 4 convolutional layers (conv) with kernel size 3×3 and stride 1. The deconvolutional layer in object coordinate regressor is 2×2 with stride 2. The last conv is 3×3 with stride 1. The final output for object coordinate is $d \times (\text{sigmoid}(p_{obj}) - 0.5)$, where d is the approximated diameter of the object and p_{obj} is the output pre-logits from the last conv.

et al. 2019 for the equivariant feature branch. We train all the heads in our model simultaneously in and end-to-end fashion with loss

$$L = L_{cls} + L_{box} + L_{mask} + L_{repro} + L_{multi} + L_{ef}. \tag{6.12}$$

The weights for these loss terms are not highly tuned, and are set equally. The network is trained for 200k iterations. The schedule for learning rate decay follows He et al. 2017. For RANSAC at the test time, the threshold for inliers is set to 1px, and number of hypotheses is 256. The refinement runs up to 100 times.

6.4 Experiments

We first introduce the creation of the dataset we used in previous section. Second, we conduct ablation studies to investigate the effect of each supervisory signal for the object coordinate head. Third, we compare the reconstruction from our object coordinate head and the classic reconstruction-based method. Finally, we run our methods on the two real world datasets: LINEMOD (Hinterstoisser et al. 2012) and Occlusion LINEMOD (Hinterstoisser et al. 2012) and compare with the state-of-the-art learning-based methods that require the 3D model in their pipeline.

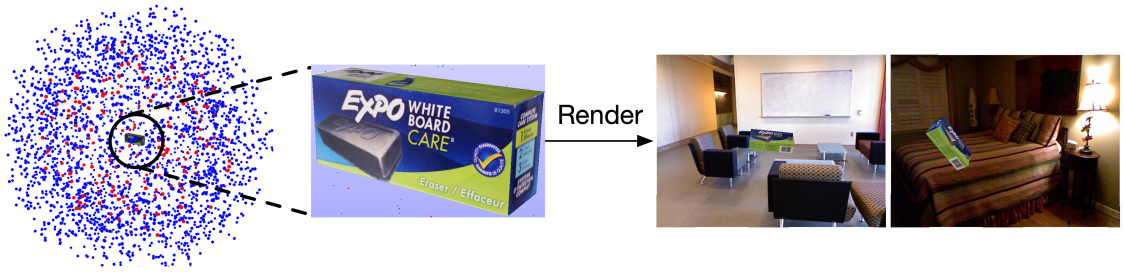


Figure 6.6: The generation of the demo synthetic dataset. Training and test viewpoints are in red and blue, respectively.

6.4.1 `Expo` Dataset

The synthetic dataset contains a square rigid object `expo`. 200 and 2500+ viewpoints are sampled from a sphere for training and test respectively. The locations of the viewpoints are randomized to make sure the object spread over the whole image frame, with various scales. We render the synthetic images using the textured CAD model from these poses. The black background is then replaced with real world images from NYU-Depth V2 Silberman et al. 2012 dataset. See figure 6.6 for examples.

6.4.2 Metric

The metrics we use to assess the pose estimation performance are ADD and 5cm5deg, same with chapter 3. ADD is the average 3D distance of model points transformed by the predicted pose and ground truth pose. For symmetric objects, ADD is relaxed to ADD-S, which is the distance between the closest points in two transformed models. If the average (or closest) distance derived by a test pose is less than 10% of the object diameter, the pose estimate is considered correct. As for 5cm5deg, an estimate is correct when the translation and rotation error is below $(5cm, 5°)$. The numbers we report in table 6.1, 6.2 and 6.3 are the proportion of frames with correct pose estimates among all test images.

	depth	repro	repro+ef	repro+ef+rgb
5cm5deg	61.3	14.3	39.3	53.1
ADD-10	57.1	23.6	52.5	58.5

Table 6.1: The pose estimation performance of different combinations of the loss terms on test set of `expo`.

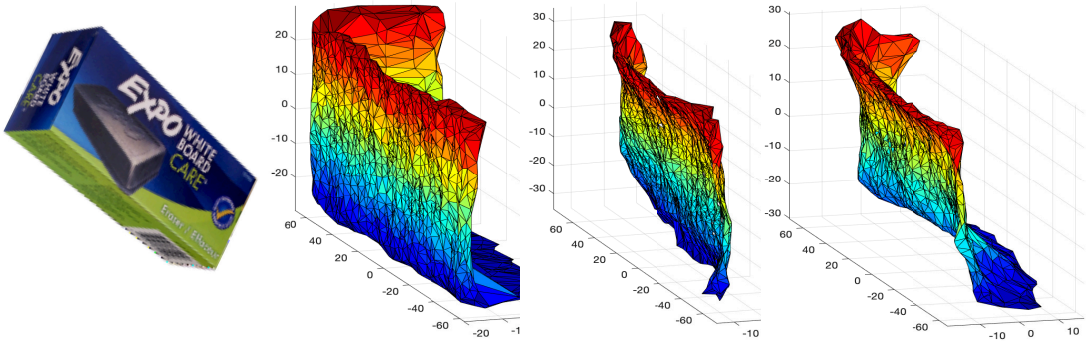


Figure 6.7: Visualization of the reconstruction from the object coordinate head. (a) is a test image. (b) is the true reconstruction from this viewpoint. (c) is the output from the head trained with reprojection loss. (d) is the output from the head trained with multi-view loss in addition to reprojection loss.

6.4.3 Ablations

We train the network using three different supervisions: (i) direct supervision from depths (as a reference); (ii) single-view reprojection loss; (iii) single-view reprojection loss along with multi-view geometry losses. The qualitative meshed visualization of the predicted 3D points from models trained with different losses is shown in figure 6.7. The quantitative results for pose estimation are shown in table 6.1.

Figure 6.7(b) shows that the true shape of the object from the test viewpoint comprises 3 perpendicular planes. With only the single-view reprojection loss as supervision, the network failed to discover the geometry of the object and predicts a set of points that lies on a plane (See figure 6.7(c)). What is interesting is that these erroneous object coordinates surprisingly result in highly (5cm5deg: 99% and ADD-10: 99.5%) accurate poses for the *training* set. It suggests that optimizing the loss term (6.5) overfits the model to produce an arbitrary shape, as long as whose projections from the ground truth pose match the silhouette of the object on the

image. Hence the correspondences built by this shape and the 2D positions result in fine pose estimates for training set (the performance on test set is reported later). In contrast, the reconstruction from the model trained with additional multi-view losses shows the corner and the 3-face structure of the object in figure 6.7(d). Quantitatively, the median chamfer distances (two-way, in m, smaller is better) between single-view reconstruction against the groundtruth shape are (0.152, 0.067), and for multi-view reconstruction they are (0.094, 0.048).

The failure caused by using reprojection loss as the only supervision also presents in the quantitative results for the test images. In table 6.1(`repro`), the 5cm5deg and ADD-10 accuracy for the model trained with reprojection loss are only 14.3% and 23.6%. This is because that the trained model dose not encode the true geometry and therefore generalizes poorly to the unseen images.

In column `repro+ef`, the model is trained with reprojection loss and equivariant feature alignment loss. The accuracy increases to 39.3% (5cm5deg) and 52.5% (ADD-10), which is approximately 2.5 times of `repro`. The best performance comes from the column `repro+ef+rgb`. It is achieved by training the model with reprojection loss and all multi-view losses ($L_{ef\_align} + L_{rgb}$). It shows that with additional multi-view constraints provided by the photometric loss, the object coordinate achieves a better pose estimates, which is even comparable with the model from direct supervision, whose accuracy is 61.3% (5cm5deg) and 58.5% (ADD-10).

6.4.4 Equivariant Feature Matching

We show several examples of the dense matching based on the learned equivariant features in two views from LINEMOD in figure 6.8. The positions of the matched features in the source and target images are used to reconstruct the source image. These middle warped images show that the learnt features successfully build 2D-2D correspondences in two images which could be used to triangulate the coordinates of the object points in 3D.

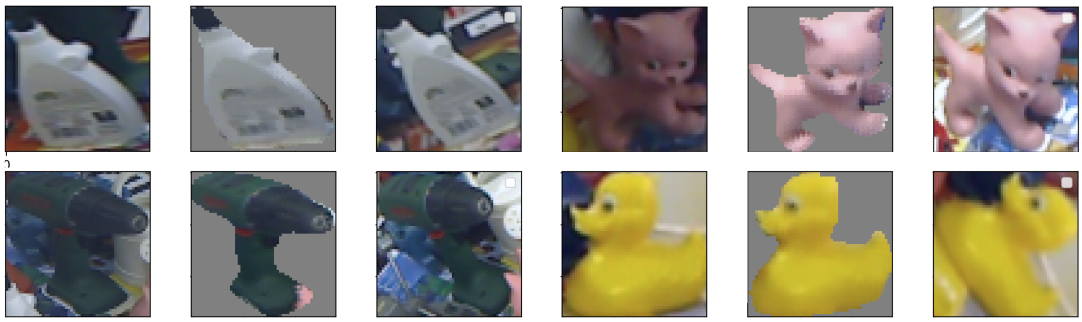


Figure 6.8: The reconstruction (middle) of the source (left) by warping the target (right) using matched feature positions.

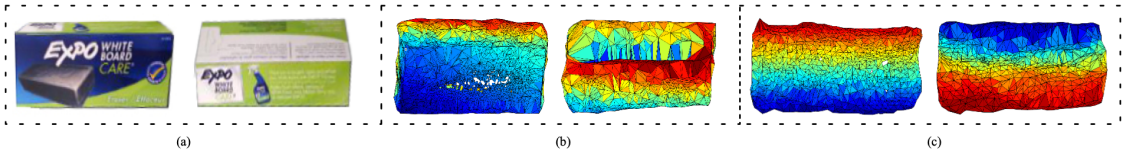


Figure 6.9: Comparison between the reconstructions from SfM and our method. Left: images from two example viewpoints; Middle: meshed reconstructions from SfM; Right: meshed reconstructions from our model.

6.4.4.1 Comparison with SfM-based Method

We run SfM using colmap (Schönberger et al. 2016; Schonberger et al. 2016) from 200 training images in `expo` datasets to build an explicit reconstruction from the sparse features. Figure 6.9 compares the reconstruction from SfM and our object coordinate head. It shows that only five out of the six planes of the object are successfully built by SfM. Apparently it is caused by the lack of textures on the missing plane, where the sparse feature detector struggles to recognize any salient points. In contrast, our model manages to build every surface despite its textureness. Our hypothetical explanation is that the backbone explores both coarse and fine features from multiple scales therefore it is more robust to the density of the visual features on the image.

6.4.5 Pose Results

6.4.5.1 On LINEMOD

We train our network strictly following the training/test split in Tekin et al. 2018. No additional synthetic data is required, as well as the 3D CAD model in our

	w/ CAD model								w/o CAD model		
method	BB8[a]	BB8 w/ r	SSD-6D[b] w/ r	Tekin[c]	DeepIM[d] w/ r	Dense-Fusion[e]	Pix2-Pose[f]	PVNet[g] w/ r	SSD-6D	chapter 3	Ours
ape	27.9	40.4	65	21.62	77.0	92	58.1	43.62	0.00	38.8	52.91
benchwise	62.0	91.8	80	81.80	97.5	93	91.0	99.90	0.18	71.2	96.51
cam	40.1	55.7	78	36.57	93.5	94	60.0	86.86	0.41	52.5	87.84
can	48.1	64.1	86	68.80	96.5	93	84.4	95.47	1.35	86.1	86.81
cat	45.2	62.6	70	41.82	82.1	97	65.0	79.34	0.51	66.2	67.30
driller	58.6	74.4	73	63.51	95.0	87	76.3	96.43	2.58	82.3	88.70
duck	32.8	44.3	66	27.23	77.7	92	43.8	52.58	0.00	32.5	54.74
eggbox*	40.0	57.8	100	69.58	97.1	100	96.8	99.15	8.90	79.4	94.74
glue*	27.0	41.2	100	80.02	99.4	100	79.4	95.66	0.00	63.7	91.98
holepuncher	42.4	67.2	49	42.63	52.8	92	74.8	81.92	0.30	56.4	75.41
iron	67.0	84.7	78	74.97	98.3	97	83.4	98.88	8.86	65.1	94.59
lamp	39.9	76.5	73	71.11	97.5	95	82.0	99.33	8.20	89.4	96.64
phone	35.2	54.0	79	47.74	87.7	93	45.0	92.41	0.18	65.0	89.24
average	43.6	62.7	79	55.95	**88.6**	**94**	72.4	**86.27**	2.42	65.2	**82.88**

Table 6.2: LINEMOD: Percentages of correct pose estimates in ADD-10. We highlight the methods whose average accuracy is above 80%. * denotes that the object is symmetric and is evaluated in ADD-S. w/r means the pose is refined with 3D model.

[a] Rad et al. 2017.
[b] Kehl et al. 2017.
[c] Tekin et al. 2018.
[d] Li et al. 2018.
[e] Wang et al. 2019.
[f] Park et al. 2019.
[g] Peng et al. 2019.

	Tekin	Pose-CNN	Oberweger	PVNet	Pix2Pose	Ours
ape	2.48	9.6	17.6	15.8	22.0	7.1
can	17.48	45.2	59.3	63.3	44.7	40.6
cat	0.67	0.93	3.31	16.7	22.7	15.6
driller	7.66	41.4	62.4	25.2	44.7	43.9
duck	1.14	19.6	19.2	65.7	15.0	12.9
eggbox*	-	22.0	25.9	50.1	25.2	46.4
glue*	10.08	38.5	39.6	49.6	32.4	51.7
holepuncher	5.54	22.1	21.3	39.7	49.5	24.5
average	6.42	24.9	30.4	40.8	32.0	30.3

Table 6.3: Results on Occlusion LINEMOD. Note that all the methods requires the 3D model in the pipeline except ours. Tekin: Tekin et al. 2018, Pose-CNN: Xiang et al. 2017, Oberweger: Oberweger et al. 2018, PVNet: Peng et al. 2019, Pix2Pose: Park et al. 2019

method. We report the performance in table 6.2. Our method outperforms more than half of the learning-based methods and achieves comparable result with the state-of-the-art method, which use a large amount of synthetic training images from new viewpoints (Peng et al. 2019) and/or 3D model for refinement (Wang et al. 2019; Li et al. 2018).

6.4.5.2 On Occlusion LINEMOD

We also test our approach on a more challenging dataset: Occlusion LINEMOD, a sequence with annotations for occluded objects. ADD-10 results are shown in table 6.3 following the test scheme of Peng et al. 2019. It shows the robustness of our method to occlusion.

6.4.5.3 On YCB-Video

We train our network using 20% images of each sequence, and visualize the results on samples of the rest. These result images show that our method also works for multi-object pose estimation. Thanks to the nature of Mask R-CNN, the extended model is able to detect, segment and estimate the pose for multiple objects at the same time, in a single forward. The visualization contains the results of detection(in bounding boxes), segmentation(in contours), and pose estimation. We projection

Figure 6.10: The results of our method on YCB-Video Dataset. The projected ground truth 3D point clouds of interest objects using the predicted pose estimates are consistent with the silhouettes of them, which verifies the accuracy of our 6dof pose estimates.

the 3D points of the ground truth object model using the predicted poses. The shape of the predicted projections match up with the silhouettes of the objects.

6.5 Conclusion

We have proposed a method that performs accurate 6dof object pose estimation from a single RGB image. Our learning-based method implicitly encodes the object

reconstruction into a network by regressing object pixel to 3D object coordinate. It then carries out 2D-3D correspondences for geometric pose solving at inference time. The learning of the network explicitly enforces the multi-view geometric constraints for the object coordinates. The additional equivariant feature branch provides consistence for objects across multiple views. We explore self-supervision for learning from image deformation and eliminates the need of 3D model in the system. Our 3D model free method reduced the performance gap between approaches with and without 3D model.

Bibliography

Wagner, Daniel, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg (2008). "Pose tracking from natural features on mobile phones". In: *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE, pp. 125–134.

Pan, Qi, Gerhard Reitmayr, Edward Rosten, and Tom Drummond (2010). "Rapid 3D modelling from live video". In: *The 33rd International Convention MIPRO*. IEEE, pp. 252–257.

Pan, Qi, Gerhard Reitmayr, and Tom Drummond (2009). "ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition." In: *BMVC*. Vol. 2. Citeseer, p. 6.

Dementhon, Daniel F and Larry S Davis (1995). "Model-based object pose in 25 lines of code". In: *International journal of computer vision* 15.1-2, pp. 123–141.

Zhang, Zhengyou (1994). "Iterative point matching for registration of free-form curves and surfaces". In: *International journal of computer vision* 13.2, pp. 119–152.

Marchand, Eric, Patrick Bouthemy, François Chaumette, and Valérie Moreau (1999). "Robust real-time visual tracking using a 2D-3D model-based approach". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 1. IEEE, pp. 262–268.

Lepetit, Vincent, Pascal Fua, et al. (2005). "Monocular model-based 3d tracking of rigid objects: A survey". In: *Foundations and Trends® in Computer Graphics and Vision* 1.1, pp. 1–89.

Pauwels, Karl, Leonardo Rubio, Javier Diaz, and Eduardo Ros (2013). "Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2347–2354.

Xiang, Yu, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox (2017). "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes". In: *arXiv preprint arXiv:1711.00199*.

Li, Yi, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox (2018). "Deepim: Deep iterative matching for 6d pose estimation". In: *Proceedings of the European Conference on Computer Vision*, pp. 683–698.

Sundermeyer, Martin, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel (2018). "Implicit 3d orientation learning for 6d object detection from rgb images". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 699–715.

Hu, Yinlin, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann (2019). "Segmentation-driven 6d object pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3385–3394.

Peng, Sida, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao (2019). "Pvnet: Pixel-wise voting network for 6dof pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4561–4570.

Wang, Chen, Danfei Xu, Yuke Zhu, Roberto Martin-Martin, Cewu Lu, Li Fei-Fei, and Silvio Savarese (2019). "DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Xiao, Yang, Xuchong Qiu, Pierre-Alain Langlois, Mathieu Aubry, and Renaud Marlet (2019). "Pose from Shape: Deep Pose Estimation for Arbitrary 3D Objects". In: *arXiv preprint arXiv:1906.05105*.

Bui, Mai, Sergey Zakharov, Shadi Albarqouni, Slobodan Ilic, and Nassir Navab (2018). "When regression meets manifold learning for object recognition and pose estimation". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1–7.

Park, Kiru, Timothy Patten, and Markus Vincze (2019). "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7668–7677.

Hinterstoisser, Stefan, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab (2012). "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes". In: *Asian conference on computer vision*. Springer, pp. 548–562.

Brachmann, Eric, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother (2014). "Learning 6d object pose estimation using 3d object coordinates". In: *European conference on computer vision*. Springer, pp. 536–551.

Brachmann, Eric, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother (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*, pp. 3364–3372.

Kehl, Wadim, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab (2017). "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1521–1529.

Chang, Angel X, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. (2015). "Shapenet: An information-rich 3d model repository". In: *arXiv preprint arXiv:1512.03012*.

Rad, Mahdi and Vincent Lepetit (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*, pp. 3828–3836.

Agarwal, Sameer, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski (2011). "Building rome in a day". In: *Communications of the ACM* 54.10, pp. 105–112.

Kendall, Alex, Matthew Grimes, and Roberto Cipolla (2015). "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946.

Brachmann, Eric, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother (2017). "Dsac-differentiable ransac for camera localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6684–6692.

Brachmann, Eric and Carsten Rother (2018). "Learning less is more-6d camera localization via 3d surface regression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4654–4662.

Brahmbhatt, Samarth, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz (2018). "Geometry-aware learning of maps for camera localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625.

Fischler, Martin A and Robert C Bolles (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6, pp. 381–395.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Thewlis, James, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi (2019). "Unsupervised Learning of Landmarks by Descriptor Vector Exchange". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6361–6371.

Oberweger, Markus, Mahdi Rad, and Vincent Lepetit (2018). "Making deep heatmaps robust to partial occlusions for 3d object pose estimation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 119–134.

Tekin, Bugra, Sudipta N Sinha, and Pascal Fua (2018). "Real-time seamless single shot 6d object pose prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 292–301.

Rad, Mahdi, Markus Oberweger, and Vincent Lepetit (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*, pp. 4663–4672.

Kendall, Alex and Roberto Cipolla (2016). "Modelling uncertainty in deep learning for camera relocalization". In: *2016 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, pp. 4762–4769.

Hartley, Richard and Andrew Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Thewlis, James, Hakan Bilen, and Andrea Vedaldi (2017). "Unsupervised learning of object landmarks by factorized spatial embeddings". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5916–5925.

Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie (2017). "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125.

Silberman, Nathan, Derek Hoiem, Pushmeet Kohli, and Rob Fergus (2012). "Indoor segmentation and support inference from rgbd images". In: *European conference on computer vision*. Springer, pp. 746–760.

Schönberger, Johannes L, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys (2016). "Pixelwise view selection for unstructured multi-view stereo". In: *European Conference on Computer Vision*. Springer, pp. 501–518.

Schonberger, Johannes L and Jan-Michael Frahm (2016). "Structure-from-motion revisited". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113.

7 Conclusion

Contents

This chapter summaries the material covered in our thesis and discusses possible future work in this area.

7.1 Summary of the Thesis

In this thesis, we have considered the problem of 6dof pose estimation for camera and objects using deep learning from only RGB inputs. Assuming that the training data of the scene or interest objects that contains images and their annotated poses are given, the objective of this problem is to determine the full 6dof pose of a query image that is from the same scene with a novel pose, or captures the same object from a novel viewpoint.

Our thesis addresses some of the problems encountered in previous pose estimation systems. We specifically addressed the issues of end-to-end object pose

estimation, uncertainty in pose regression, geometry-driven self-supervision of reconstruction for pose estimation (both for scenes and objects). They are all solved on the CNN base and integrated deeply with geometry. We believe our proposed methods on these issues contribute to existing approaches and made them the more completed solution to real world applications.

In chapter 3 we presented an end-to-end architecture that simultaneously detects, segments and regresses the 6dof poses of objects from a single RGB image. This is achieved by extending the recent Mask R-CNN (He et al. 2017) architecture with a new pose regression head, which directly maps the RoI features of object candidates to their poses. The output target of our pose regression branch consists of a full rotation vector and one component of translation. We used Lie algebra to represent the rotation to overcome the inherent issues of Euler angle, rotation matrix and quaternion. The full translation is derived from the z component given by our pose head and the position of bounding box from detection head. We show that our method outperforms most of the existing RGB-based methods when post-refinements is dropped, which always requires the 3D CAD model of the objects. Our method also conducts a very fast inference and meets the requirement for real time application.

In chapter 4, we presented a method that estimates the uncertainty of camera pose estimate regressed from a deep neural network. It is done by combining the deterministic CNNs and probabilistic Gaussian Process Regression into an unified, end-to-end framework. It not only directly maps the input image to 6dof pose space, but also produces the predictive distribution over the rotation and translation. The uncertainty of the pose is predicted by using Coregionalization kernel on translation and rotation vector. The input to the kernel function is the parametric pseudo inducing feature vectors learned by the network and that of the query image. We also replaced the traditional L2 norm loss based linear regressor with the KL divergence between exact posterior and the variational distribution. We show that our proposed hybrid framework improves the system efficiency of

method based on Bayesian approximate CNN, while without losing accuracy. The uncertainty of our model indicates the confidence of the pose prediction also can be interpreted as a measure of how certain the image comes from a specific scene.

In chapter 5, we proposed a reconstruction-based method for accurate camera relocalisation. It solves the 6dof pose of the camera based on the learned scene coordinates from RGB images. In order to learn the network that performs scene coordinate regression without using the ground truth 3D model of scenes, we explicitly enforce constraints for scene coordinates using multi-view geometry in a self-supervised manner. We built this constraint based on photometric consistency for the regressed scene coordinates on a pair of images from different viewpoints. We also explored the consistency in dense deep feature space, proposing a featuremetric loss that help the photometric loss when it struggles in texture-less regions. These constraints imposed by our proposed loss not only improve the efficiency of training, but also help the learnt model to produce more reliable 2D-3D correspondences which improves the camera relocalisation accuracy subsequently.

In chapter 6, we extended the proposed reconstruction-based pose estimator in chapter 5 to object-level. The process of object reconstruction is implicitly encoded by our network which regresses object pixels to their 3D object-centric coordinates. The network is achieved by building an object coordinate regression head on the base of Mask R-CNN (He et al. 2017). We deployed two branches in the new head: the object coordinate branch that predicts the geometric object coordinates from RoI features, and the equivariant feature branch which finds consistency on the surface of objects in their appearance. We matched the learnt equivariant features of object in different viewpoints to establish 2D-2D correspondences between image pairs of an object, and built multi-view constraints for the object coordinate head during learning. Using the idea of self-supervision which we introduced in chapter 5, we removed the reliance on the 3D CAD object models that have been used in methods extensively for pose estimation in training. At inference time, 2D-3D

correspondences are established by the learnt object coordinate branch, and used to solve the 6dof object pose. The competitive results of our methods shows that our 3D model free method closed the performance gap between approaches using and not using 3D model.

7.2 Some Insights

From the aforementioned approaches, one can see that two families of methods are adopted in this thesis, which ar regression-based and reconstruction-based methods. Notwithstanding that we manage to bring uncertainty into the regression-method, the accuracy performance of pose estimation (especially camera relocalisation) is much poorer than what is achieved in reconstruction-based method. Again, a question raises up that: Are CNNs in nature good at modeling relationship (in the case of this thesis is pose) *directly*?

Unfortunately experiment results empirically suggest that they are not, especially when there is a strict constraints on the form of the representation of the relationship, such as the rotation component in the 6dof pose.

We hypothetically give three speculations to the reason below:

- **The spatial arrangement of the image details are omitted.** In a conventional CNNs architecture which performs image-level regression or classification, pooling layers and/or fully connected layers are commonly used to obtain features with invariance to spatial particularities of image patches. However, the position and orientation of these details is fundamentally crucial to the geometric relationship therefore should not be ignored during the abstraction.

- **The representation of the pose is strongly non-linear.** It is well-known that machine learning methods are in nature have a flair for interpretation

and generalization in a linear output space. However, the geometric representation of pose is in a non-linear space, such as the orthogonality in the rotation matrix and the unity in the quaternion. In contrast, the output of reconstruction-based methods are coordinates in a linear Euclidean space, therefore the networks do not struggle from interpretation as was shown in then reconstruction-based methods in chapter 5 and 6.

- **Relationship involves two parties.** The modelling of relationship requires two identities. In widely applied CNNs, the tasks commonly only focus on the being of one of them, such as the objects (what are they, where are they in the image), and less attention is paid to the state of the other identity, for instance the camera (such as where are the objects to the camera). Methods using the pre-defined (or reconstructed) objects, which essentially provide full information about one of the identities, are able to show effectiveness on estimating the state of the camera, as shown in the reconstructed-based methods for estimating object pose – which is essentially the relationship between objects and camera.

With these bear in mind, we suggest that in a relationship prediction task, instead of directly regressing to relationship using a deep model, the network should be considered as a tool for building connections between identities based on an intermediate representation. The relationship then can be solved according to these connections, indirectly.

7.3 Future Work

Though the techniques presented in this thesis improve the capability of 6dof pose estimation compared to earlier solutions, much work still remains to make these systems suitable for requirements outside the lab. In this section, the improvements

that could be made to our proposed techniques will be discussed, as well as more general areas for future research.

First, for the uncertainty of a deep network prediction, especially from the framework we proposed in chapter 4, we would like to explore their potential applications in other established real world systems. It will be interesting to see how does the uncertainty work in a robotics system, such as the probabilistic visual SLAM. The uncertainty of the pose of the robots and the landmarks in the online build map are crucial for state estimation. For instance, in the traditional filter-based state estimator used in visual SLAM, a motion model of the robot is formulated to predict the next state of robot based on current state, whereas a measurement model provides the measurement from current state. They are recursively fused together to find the best estimate of the state. The noise of the states and measurement are always assumed as zero-mean Gaussian during the initialization of the whole system. If our proposed method is integrated to a visual SLAM system, it not only estimates an absolute pose to correct the incremental pose tracker, but also provides a more realistic pose uncertainty, instead of using naive white noise.

Second, the reconstruction-based network for camera relocalisation that we proposed in chapter 5 is scene-specific, which means we have to train specific models to different scenes. A valuable direction of research is to train a general model that works for any scenes with similar statistics, arrangement or scale. One can of course easily try to use all training images in the dataset from different scenes to train a single network. Most likely it will work on the test images from the original datasets, but a more significant question we should ask is that dose it generalize to new scenes. Intuitively, this seems not possible to be achieved since the nature of the problem is *re*-localisaion. But we can adaptively use the idea of world-centric reconstruction and pose estimation to the problem of camera-centric reconstruction, which is basically the depth estimation problem (and the depth is fundamentally

predicted mainly relying the image statistics). Therefore, we believe it is worth to try to apply the philosophy of learning-based reconstruction to depth estimation.

As for object pose estimation, we also would like to pursue the ability of generalization to object class. The task we completed in chapter 3 and chapter 6 treats each object as a class with an individual model that is very specific to the object itself. It however limits the generalization of the method to different objects from the same class. It has been shown by Suwajanakorn et al. 2018; Thewlis et al. 2019 that 3D structure represented by 3D keypoints (Suwajanakorn et al. 2018), and 2D landmarks (Thewlis et al. 2019) can be learned to generalize to the category level of object. It means that practically it is feasible to learn the framework we presented in chapter 6 to build a sparse reconstruction for the objects from the same class, which subsequently can be used to perform 6dof pose estimation. The prerequisite is to find an unified pose representation for all objects in the class to enable the training of the network. Such representation has to be consistent to all instances and able to conduct the projection of general 3D keypoints to the 2D image plane (or implicitly find the connection between them).

Third, one limitation of our methods is that we treat camera pose estimation and object pose estimate separately as two independent tasks. In other words, moving objects are ignored in camera relocalisation. However in a real world application, the movement of camera and object both are the interest of motion estimation. An example is that when the car moves in an urban environment and localize itself in the global map, other traffic participants such as pedestrians and cars will simultaneously move with respect to the host car. Therefore, it is worth trying to combine the proposed methods into a unified system, building a motion segmentation method that performs two tasks at the same time, using image from the same source.

At last, we also think it will be exciting to see more research will be conducted that aim to solve the problem of 6dof pose estimation for object with non-trivial

characteristics, such as geometric symmetry and non-rigidity, using the technique of deep learning.

Bibliography

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). "Mask r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Suwajanakorn, Supasorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi (2018). "Discovery of latent 3d keypoints via end-to-end geometric reasoning". In: *Advances in Neural Information Processing Systems*, pp. 2059–2070.

Thewlis, James, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi (2019). "Unsupervised Learning of Landmarks by Descriptor Vector Exchange". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6361–6371.