UNIVERSITY OF TECHNOLOGY SYDNEY

DOCTORAL THESIS

Action for Perception: Active Object Recognition and Pose Estimation in Cluttered Environments

Kanzhi WU

A thesis submitted in fulfilment of the requirements

for the degree of Doctor of Philosophy

at the

Centre for Autonomous Systems

School of Electrical, Mechanical and Mechatronic Systems

August 8, 2017

Certificate of Original Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as part of the collaborative doctoral degree and/or fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student:

____

Date:

____

Abstract

Object recognition and localisation are indispensable competency for service robots in everyday environments like offices and kitchens. Presence of similar objects that can only be differentiated from a small part of the surface together with clutter that leads to occlusions make it impossible to detect target objects accurately and reliably from a single observation. When the sensor observing the environment is mounted on a mobile platform, object detection and pose estimation can be facilitated by observing the environment from a series of different viewpoints. Computing *Active perception* strategies, with the aim of finding optimal actions to enhance object recognition and pose estimation performance is the focus of this thesis.

This thesis consists of two main parts:

In the **first** part, it focuses on object detection and pose estimation from a single frame of observation. Using an RGB-D sensor, we propose a modular 3D textured object detection and pose estimation framework which can recognise object under cluttered environment by taking advantage of the geometric information provided from the sensor. To handle less-textured objects and objects under severe illumination conditions, we propose a novel RGB-D feature which is robust to illumination, scale, rotation and viewpoint variations, and provides reliable feature matching results under challenging conditions. The proposed feature is validated for multiple applications including object detection and point cloud alignment. Parts of the above approaches are integrated with existing work to produce a practical and effective perception module for a warehouse automation task. The designed perception system can detect objects of different types and estimate their poses robustly thus guaranteeing a reliable object grasping and manipulation performances.

In the **second** part of the thesis, we investigate the problem of *active* object detection and pose estimation from two perspectives: with and without considering the uncertainties in the motion model and the observation model. First, we propose a model-driven active object recognition and pose estimation system via exploiting the feature association probability under scale and viewpoint variations. By explicitly modelling the feature association, the proposed system can predict future information more accurately thus laying the foundation of a successful active Next-Best-View planning system even with a naive greedy search technique. We also present a probabilistic framework which handles motion and observation uncertainties in the active object detection and pose estimation problem. We present an optimisation framework which computes the optimal control at each step, using an objective function which incorporates uncertainties in state estimation, feature coverage for better recognition confidence and control consumption. The proposed framework can handle various issues such as object initialisation, collision avoidance, occlusion and changing the object hypothesis. Validations based on a simulation environment are also presented.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Gamini Dissanayake for his continuous support during my PhD study, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my PhD study. I would also like to thank my co-supervisor Dr. Ravindra Ranasinghe, the door of his office was always open whenever I ran into a trouble spot or had a question about my research or writing.

My sincere thanks also go to Assoc. Prof. Shoudong Huang and Mr Teng Zhang for the insightful discussions about SLAM and planning under uncertainty. During the past four years, I have experienced a welcoming and passionate research atmosphere in CAS and I would like to thank Prof. Dikai Liu for his management of CAS. I would like to thank Dr. Liang Zhao for patiently answering my questions about multiview geometry.

I would also like to thank my great friends and colleagues in CAS: Daobilige Su, Lakshitha Dantanarayana, Kasra Khosoussi, Raphael Falque and so many other colleagues who have helped me, too many to list here. The discussions with you not only have helped me in my academic research but have also enriched my life in the past four years.

Last but not the least, I would like to thank my parents and especially my wife Liye Sun for her understanding, support and companionship during my research career.

Contents

List of Figures

List of Tables

List of Abbreviations

ANN	**A**pproximate **N**earest **N**eighbour
APC	**A**mazon **P**icking **C**hallenge
BoW	**B**ag **o**f **W**ords
BRAND	**B**inary **R**obust **A**ppearance and **N**ormals **D**escriptor
BRIEF	**B**inary **R**obust **I**ndependent **E**lementary **F**eatures
BRISK	**B**inary **R**obust **I**nvariant **S**calable **K**eypoints
BRM	**B**elief **R**oad**M**ap
CNN	**C**onvolutional **N**eural **N**etworks
DLT	**D**irect **L**inear **T**ransformation
DoF	**D**egree of **F**reedom
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
FIRM	**F**eedback-based **I**nformation **R**oad**M**ap
GLOH	**G**radient **L**ocation and **O**rientation **H**istogram
GBS	**G**eneral **B**elief **S**pace
ISS	**I**ntrinsic **S**hape **S**ignatures
KPQ	**K**ey**P**oint **Q**uality
LBSS	**L**aplace-**B**eltrami **S**cale-**S**pace
LDA	**L**inear **D**iscriminant **A**nalysis
LLC	**L**ocality-constrainted **L**inear **C**oding
LM	**L**evenburg **M**arquart
LQG	**L**inear **Q**uadratic **G**aussian
LQR	**L**inear **Q**uadratic **R**egulator
LOIND	**L**ocal **O**rdinal **I**ntensity and **N**ormal **D**escriptor
LSH	**L**ocal **S**ensitive **H**ashing
MDP	**M**arkov **D**ecision **P**rocess
MOPED	**M**ultiple **O**bject **P**ose **E**stimation and **D**etection

MPC	**M**odel **P**redictive **C**ontrol
NBV	**N**ext-**B**est-**V**iew
ORB	**O**riented FAST and **R**otated **B**RIEF
PCA	**P**rinciple **C**omponent **A**nalysis
PCL	**P**oint **C**loud **L**ibrary
PFH	**P**oint **F**eature **H**istogram
PnP	**P**erspective-**n**-**P**oint
POMDP	**P**artially **O**bservable **M**arkov **D**ecision **P**rocess
PRM	**P**robabilistic **R**oad**M**ap
RANSAC	**RAN**dom **SA**mple **C**onsensus
RISAS	A **R**otation, **I**llumination and **S**cale invariant **A**ppearance and **S**hape Feature
R-CNN	**R**egion-based **C**onvolutional **N**eural **N**etworks
RNN	**R**esidual **N**eural **N**etwork
ROI	**R**egion **O**f **I**nterest
RRG	**R**apid-exploring **R**andomised **G**raph
RRT	**R**apid-exploring **R**andomised **T**rees
RRBT	**R**apid-exploring **R**andomised **B**elief **T**rees
SfM	**S**tructure from **M**otion
SHOT	**S**ignature of **H**istogram **O**rien**T**ation
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalisation **A**nd **M**apping
SORSAC	**S**patially **OR**dered **SA**mple **C**onsensus
SQP	**S**equential **Q**uadratic **P**rogramming
SURF	**S**peed-**U**p **R**obust **F**eature
SVD	**S**ingular **V**alue **D**ecomposition
RANSAC	**RA**ndom **SA**mple **C**onsensus
USC	**U**nique **S**hape **C**ontext

Chapter 1

Introduction

1.1 Background and Motivation

Autonomous mobile service robots working in close cooperation with humans have a great potential to help individuals with special needs for improving their lives. These robots will have the technology to support the autonomy and independence of individuals with special needs. 3D object recognition and pose estimation (also known as object localisation) is an indispensable task of these service robots that enables robots to dynamically perform grasping and manipulation tasks. In order to accomplish the manipulation and interactive tasks reliably, the object detection solution for these service robots requires estimating the 6-DoF relative pose of the identified target object with respect to(w.r.t) the world coordinate frame or the sensor coordinate frame. Detecting and localising objects in everyday environments like offices and kitchens typically full of confronting conditions such as the presence of occlusion and ambiguity is very challenging.

In order to recognize objects with 6-DoF relative pose, we need strategies which are different to the object detection solutions that are commonly used in computer vision which generally provide 2D or 3D bounding-boxes[50, 153, 97, 58]. The recent emergence of deep learning[89, 22] significantly improves the performances of state-of-the-art object detection algorithms[57, 56, 120] on public dataset[83]. In published deep leaning based approaches, there have not been any provision to formulate the 6-DoFs pose estimation problem in a neural network framework until the recent work presented in [105, 78]. Different from the mentioned approaches above, traditional point feature based 3D object recognition systems[31] can provide full 6 DoFs estimate of the object pose. In the past decade, a considerable amount of work has been accomplished along with the development of novel local features[96, 17], more reliable feature matching methods[102, 10] and more accurate pose estimation algorithms[3, 8]. However, the 3D object recognition and pose estimation problem is far from solved in its most general form.

The work in this thesis focuses on the traditional object recognition and pose estimation for robotic perception tasks. In this domain, there are several well-known limitations such as occlusion, less-textured object recognition and object recognition under different illumination conditions that greatly influence the performance of the traditional object recognition algorithms. Due to scene complexity, useful information may be occluded causing ambiguity in feature correspondences leading to incorrect object recognition results. Also the limitation in illumination invariance of well-known features such as SIFT[96] and SURF[18] makes the object recognition even harder as the feature matching under extreme illumination conditions is infeasible. Most of the existing object recognition and pose estimation methods are constrained by traditional sensors such as monocular cameras. However, the availability of advanced sensors such as RGB-D cameras facilitates the capture of not only the appearance information but also the geometric knowledge of the environment. This opens up avenues to use multi-modal sensor information either to further improve the performance of the current approaches or to develop new strategies.

As mentioned earlier, most of the work in object detection has been based on a single image and the recognition performance and the accuracy of pose estimation is limited by occlusions and ambiguity in appearance and geometry. A popular approach to address this ambiguity and occlusion issue in object recognition and pose estimation is to observe the object from multiple viewpoints by moving the sensor.Through acquiring information from multiple viewpoints, previously hidden information can be discovered and distinctive features can be captured, thus enhancing the object detection confidences and the pose estimation accuracy. Planning a trajectory or a viewpoint which can maximise the performance the object detection and pose estimation framework is one key focus of this thesis. This *active sensing* was first introduced and defined as below by Bajcsy in 1988[14].

> *Active sensing is the problem of intelligent control strategies applied to the data acquisition process which will depend on the current state of data interpretation including recognition.*

Active object detection and pose estimation, as a subset of the *active sensing* problem, is targeted at finding an optimal viewpoint or series of viewpoints which enables better identification of the object and provides more accurate estimation. This problem is also known as one example of *Next-Best-View*(NBV) problems in the computer vision community[113, 34, 156] where motion uncertainty is not considered in general.

For the active object recognition and pose estimation problem, depending on whether the observation and motion uncertainties are considered, the existing approaches are categorised into two types: 1) NBV based methods[125] where the motion uncertainty is not taken into account and 2) *planning under uncertainty* based methods[149, 70] where all uncertainties are modeled and embedded in the estimation process. In the literature, most of the contributions on NBV for object recognition are focused on either developing novel representation schemes of the objects and the environments[27] or designing ambiguity functions which need to be optimised using criteria such entropy[42]. However, the potential of using information from object models has not been fully exploited and utilised to control the motion of the sensor. One of the key emphases of this thesis is to solve the active object recognition problem by constructing more information-rich object models. NBV problem has been investigated thoroughly in the field of computer vision while planning under uncertainty is a popular research topic in robotics. Various kinds of robotic applications can be formulated as planning under uncertainty problem such as exploration and active Simultaneous Localisation and Mapping (SLAM). The approaches can be further categorised into: *sampling-based approaches* and *optimisation-based approaches* and great contributions have been accomplished in the recent years such as Rapid-Exploring Random Belief Tree (RRBT), Feedback-based Information RoadMap (FIRM) and General Belief Space (GBS) planning. However, there is not any mature solution specifically designed for active object detection and pose estimation problems under the framework of planning under uncertainty; This is one particular contribution of this work. In practice, the contributions presented in this thesis are targeted at developing an autonomous perception system for mobile robotics in indoor environments which can detect both textured and less-textured objects under challenging illumination conditions by active manoeuvring in the scenario. The system should be able to plan reasonable trajectories even when motion noises and observation noises are presented in the environment.

1.2 Contributions

The main contributions of this thesis are summarised as follows:

- Chapter. 3
  - speeds up state-of-the-art object detection and pose estimation system[32] by taking advantage of information from an RGB-D sensor in feature clustering and pose estimation;

  - presents a graph-based outlier correspondences rejection algorithm which exploits the geometric constraints between each pair of feature associations;

- Chapter. 4
  - presents a new RGB-D feature, RISAS, which is robust to viewpoint, illumination, scale and rotation variations;
  - verifies the importance of combining appearance and geometric information in both keypoint detection and feature description by comparing with state-of-the-art 2D or 3D features;
  - demonstrates the effectiveness of RISAS in detecting less-textured objects or objects under extreme illumination conditions;
- Chapter. 5
  - designs an object recognition system under warehouse automation via combining 1) RGB-D/RGB recogniser for textured objects and 2) kernel descriptor based recogniser for less-textured objects;
- Chapter. 6
  - presents a model-driven active object recognition and pose estimation for detecting textured objects under cluttered indoor environments;
  - verifies the importance of considering feature matching likelihood to predict the observability of features accurately;
  - uses additional feature weighting attribute to denote the importance of the feature w.r.t the object and enables the system to differentiate similar objects in active object recognition;
- Chapter. 7
  - presents an active object recognition and pose estimation framework in general belief space while considering motion and observation uncertainties;
  - designs a comprehensive objective function which includes state uncertainty, feature coverage and repeatability, control consumption and collision avoidance;

  - solves various practical problems in active object detection and pose estimation such as object pose and uncertainty initialisation, control initialisation for optimisation and occlusion handling.

1.3 Outline

This thesis is organised as follows:

- Chapter. 2 reviews related work in both single-view and active object detection and pose estimation;
- Chapter. 3 presents a textured object detection and pose estimation under cluttered environment using an RGB-D sensor;
- Chapter. 4 proposes a novel RGB-D feature which is robust to illumination, rotation, viewpoint and scale variations and shows the superiority of the feature in detecting objects under severe illumination conditions;
- Chapter. 5 demonstrates a perception module for object recognition under warehouse environments;
- Chapter. 6 proposes a model-driven active object detection and pose estimation system which can recognise both different and similar objects via exploiting the environment;
- Chapter. 7 proposes an active object detection and pose estimation system under general belief space considering motion and observation uncertainties;
- Chapter. 8 concludes the thesis and discusses the open challenges and our future work;
- Some proofs and derivations are illustrated in detail in Appendix. A;

To summarise, Fig. 1.1 provides an overview of the contributions in this thesis listed above: Chapter. 3 and Chapter. 4 provide object detection and pose estimation solutions to both textured and less-textured objects;Combining the solution presented in Chapter. 3 with kernel descriptor and modern machine learning algorithms, Chapter. 5 provides a practical framework to handle warehouse automation tasks such as picking objects from a bin; Chapter. 6 and Chapter. 7 present active object detection and pose estimation solutions for two different scenarios: 1) considering motion and observation noises and 2) ignoring these uncertainties and both simulations and practical experiments demostrate the effectiveness of these algorithms.

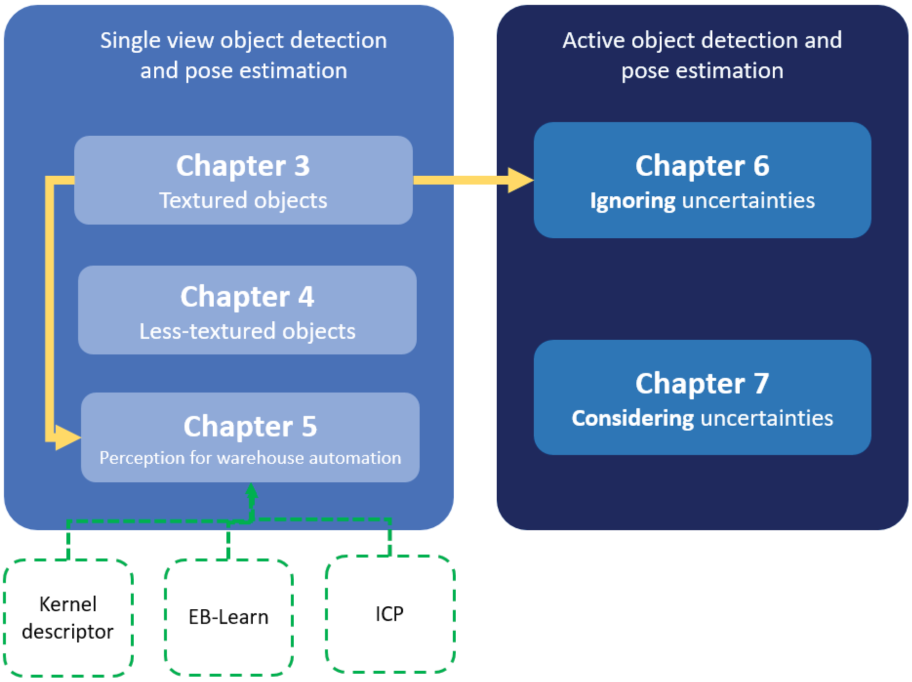

Single view object detection and pose estimation
Chapter 3
Textured objects
Chapter 4
Less-textured objects
Chapter 5
Perception for warehouse automation
Active object detection and pose estimation
Chapter 6
Ignoring uncertainties
Chapter 7
Considering uncertainties
Kernel descriptor
EB-Learn
ICP

FIGURE 1.1: Outline of the thesis and relationship between chapters.

1.4 Publication List

- **Journal *Submissions***

  1. *RISAS: A Novel Rotation, Illumination and Scale Invariant RGB-D Feature*, submitted to IEEE Transaction on Cybernetics

     Kanzhi Wu, Xiaoyang Li, Ravindra Ranasinghe, Yong Liu and Gamini Dissanayake

  2. *Convergence and Consistency Analysis for a Lie Group based 3D EKF SLAM*, IEEE Robotics and Automation Letter

     Teng Zhang, **Kanzhi Wu**, Jingwei Song, Shoudong Huang and Gamini Dissanayake

- **Conference Papers**

  1. *Planar Scan Matching Using Incident Angle*, submitted to IEEE/RSJ International Conference on Intelligent Robotis and Systems, 2017

     Jixin Lv, Yue Wang, **Kanzhi Wu**, Gamini Dissanayake, Yukunori Kobayashi and Rong Xiong

  2. *An Invariant-EKF VINS Algorithm for Improving Consistency*, submitted to IEEE/RSJ International Conference on Intelligent Robotics and Systems, 2017

     Kanzhi Wu, Teng Zhang, Daobilige Su, Shoudong Huang and Gamini Dissanayake

3. *Gyro-Aided Camera-Odometer Online Calibration and Localization*, American Control Conference, 2017

   Dongxuan Li, Kevin Eckenhoff, **Kanzhi Wu**, Yue Wang, Rong Xiong, Guoquan Huang

4. *RISAS: A Novel Rotation, Illumination, Scale Invariant Appearance and Shape Feature*, IEEE International Conference on Robotics and Automation, 2017

   Kanzhi Wu, Xiaoyang Li, Ravindra Ranasinghe, Gamini Dissanyake and Yong Liu

5. *Active Object Detection and Pose Estimation in General Belief Space*, Robotic: Science and Systems workshop on Robot-Environment Interaction for Perception and Manipulation, 2016

   Kanzhi Wu, Teng Zhang, Shoudong Huang, Gamini Dissanayake and Ravindra Ranasinghe

6. *Active Recognition and Pose Estimation of Household Objects in Clutter*, IEEE International Conference on Robotics and Automation, 2015

   Kanzhi Wu, Ravindra Ranasinghe and Gamini Dissanayake

7. *A Fast Pipeline for Textured Object Recognition in Clutter using an RGB-D Sensor*, IEEE International Conference on Control, Automation, Robotics and Vision, 2014

   Kanzhi Wu, Ravindra Ranasinghe and Gamini Dissanayake

Chapter 2

Literature Review

Object recognition and pose estimation is an indispensable task in robotic perception and manipulation. As an interdisciplinary research topic across robotics, computer vision, and machine learning, enormous progress has been made in the past few decades. In this chapter, we review some of the state-of-the-art work in 1) single view object recognition and pose estimation and 2) active object recognition and pose estimation especially the solutions which are closely related to the key contributions of this thesis. Fig. 1.1 presents a clear illustration of the structure and reviewed literature in this chapter. The blocks with red border highlight the work which is closely related with the contributions presented in following chapters. The block with dotted border is not the focus of this chapter and is only briefly introduced in this chapter.

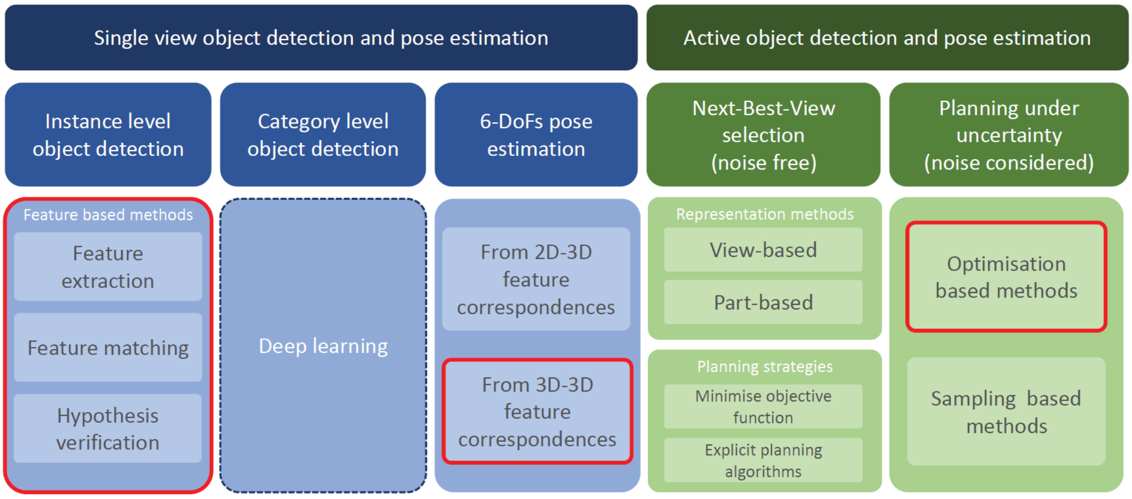
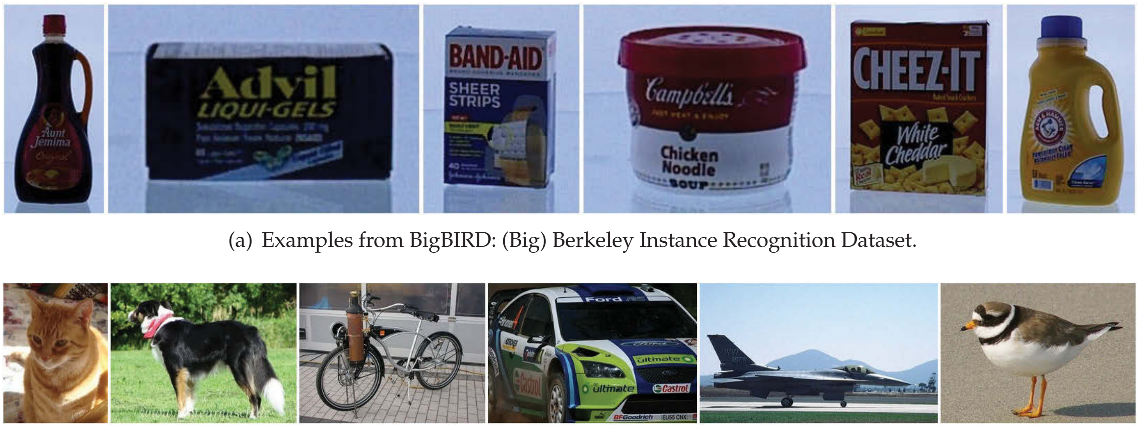

Single view object detection and pose estimation
Active object detection and pose estimation
Instance level object detection
Category level object detection
6-DoFs pose estimation
Next-Best-View selection (noise free)
Planning under uncertainty (noise considered)
Feature based methods
Feature extraction
Feature matching
Hypothesis verification
Deep learning
From 2D-3D feature correspondences
From 3D-3D feature correspondences
Representation methods
View-based
Part-based
Planning strategies
Minimise objective function
Explicit planning algorithms
Optimisation based methods
Sampling based methods

FIGURE 2.1: Outline of the reviewed literature in Chapter. 2.

2.1 Single View Object Recognition and Pose Estimation

The first two subsections present contributions about instance-level object recognition and category-level object recognition. The focus of instance-level object recognition algorithms is to recognise a specific object such as a coke bottle in the scene. On the other hand, the focus of category-level object recognition algorithms is to recognise a certain category of objects such as bottles in the scene. The approaches towards solving these two problems show remarkable differences. Local feature description, feature matching, and multi-view geometry are fundamental elements towards building instance-level object recognition systems. However, with category-level object recognition prime emphasis is given on learning a robust higher-level representation of each object category thus is involved more with machine learning techniques. Fig. 2.2 illustrates the differences between two problems using two well-known datasets, BigBIRD dataset[133] and PASCAL VOC dataset[49]. The first row shows examples from BigBIRD dataset which includes objects of specific types and the second row shows examples from VOC dataset which include objects of general categories such as dogs and cats. The last subsection discusses estimating the 6-DoFs relative pose of the objects using different sensors such as a monocular camera and an RGB-D camera.



(a) Examples from BigBIRD: (Big) Berkeley Instance Recognition Dataset.

(b) Examples from PASCAL Visual Object Classes (category-level) Dataset.

FIGURE 2.2: Differences between instance-level object recognition dataset and category-level object recognition dataset.

2.1.1 Instance-Level Object Recognition

The problem of instance-level object recognition stems from the need for recognising and verifying faces in public security applications. Typically, *global feature representation* based approaches

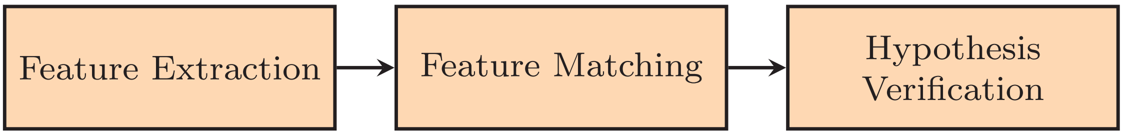


FIGURE 2.3: General pipeline for instance-level object recognition.

use the entire image to calculate the similarity between the query image and given examples(templates). Turk and Pentland[146] generated a vector space from the images and used Principal Component Analysis (PCA) to reduce the dimension of the vector space (i.e. raw image data) thus obtaining more compact encodings. Belhumeur and Kriegman[20] proposed *Fisherfaces* to optimise class separability using Fisher's Linear Discriminant Analysis (LDA) rather than the subspace generated from PCA. Despite the success in face detection, the limitations of *global feature representation* are evident. It is unable to deal with occlusion scenarios, viewpoint variations, and deformable objects. However, soon after the invention of well-performed local features, e.g, Scale Invariant Feature Transform (SIFT)[96], *local feature representation* based approaches became popular for solving the instance-level object recognition problem.

Fig. 2.3 summarises the general steps for solving the instance recognition problem. A typical instance recognition framework consists of 3 steps:

1. Extracting local features from both models and query images independently;
2. Finding the correct correspondences between extracted features;
3. Generating the object hypothesis and verifying the hypothesis w.r.t a certain criterion such as rigid transformation constraint;

This subsection firstly describes some influential work in these 3 key steps and later review recent systematic work on instance-level object recognition.

Feature Extraction: Keypoint Detection and Descriptor Construction The feature extraction problem can be further divided into two sub-problems: keypoint detection(also known as localisation) and descriptor construction. The feature extraction methods SIFT (as per Lowe's original research publications [96][1]) and Speed-Up Robust Feature(SURF)[17] are designed to consist of both detector and descriptor together. However, some feature extraction methods focus only on either keypoint detection or descriptor construction. For example, Feature from Accelerated Segment Test(FAST)[123] method focuses only on keypoint detection

[1] In some other literature[100], SIFT feature is treated as a descriptor since it can be combined with other keypoint detectors independently.

while Binary Robust Independent Elementary Feature (BRIEF)[29] focuses only on descriptor construction.

Based on the source of the sensor information where the features are extracted from, this subsection is organised into 3 parts:

- **2D Image Features**

  The research work on keypoint detectors dates back to the 1980s when Hessian detector[19] and Harris detector[62] were firstly invented. SIFT feature is one of the most attractive local features, proposed by Lowe. SIFT combines a Different-of-Gaussian interest region detector and gradient orientation histogram as the descriptor. By constructing the descriptor from a scale-normalised and orientation-normalised image patch, SIFT feature is made robust to scale and rotation variations. In order to improve the performance of SIFT, Bay et al. proposed SURF[17] an alternative which utilises a Hessian-Laplace region detector with another gradient orientation based feature descriptor. SURF adopts simple 2D box-shaped filters to approximate the derivative filter kernels which require less time for processing and hence achieving improved timing efficiency. Followed by SIFT and SURF, many other gradient histograms based local features were proposed such as PCA-SIFT from Ke and Sukthankar[77] and Gradient Location and Orientation Histogram (GLOH) from Mikolajczyk and Schimid.

  Even though GPU implementations of SIFT(GPU-SIFT[159]) and SURF(GPU-SURF[139]) are available, researchers are still focusing on designing efficient features which can be practically implemented on platforms without GPU. Rosten and Drummond proposed FAST[123] which uses a Bresenham circle of radius equal to 3 pixel to decide whether a candidate point is a keypoint or not. Several machine learning techniques are also introduced in FAST to speed up the keypoint detection and improve the repeatability. Using FAST keypoint detector, several binary local features were proposed. Calonder et al. proposed BRIEF[29] and formulated the feature descriptor as a binary string. BRIEF feature takes relatively less memory and can be matched faster using Hamming distance in real-time under limited computational resources. However, BRIEF is not designed to be robust to scale variations. Leutenegger et al. proposed Binary Robust Invariant Scalable Keypoint (BRISK)[92] which has a scale invariant keypoint detector and a binary string like descriptor. Similar to SURF and SIFT, in order to reach the scale invariant capability, points of interest are identified across multiple scales using a saliency criterion. The descriptor is constructed using a similar principle as presented in BRIEF by sampling in the neighbourhood region of the keypoint. Oriented FAST and Rotated BRIEF(ORB)[126], another well-known binary feature, has been widely used in the SLAM community[104]. ORB is built on FAST keypoint detector and BRIEF descriptor, and it is invariant to rotation variations and more robust to noise compared with BRIEF.

- **3D Geometry Local Feature**

  Based on the scale invariant capability, Tombari et al. [143] categorised 3D keypoint detectors into 2 classes: *fixed-scale* detector and *adaptive-scale* detector. In order to select the keypoints which are salient in depth image(point cloud), researchers evaluate the quality of the keypoint by adopting different criteria, e.g., the normal vector of the surface and curvature of the mesh. Zhong et al. proposed Intrinsic Shape Signature (ISS)[163] based on the eigenvalue decomposition of the scatter matrix of the points which belong to the support set of the candidature point. Another example of fixed-scale 3D keypoint detector is KeyPoint Quality (KPQ) proposed by Mian et al[99]. Similar to ISS, KPQ is also based on the scatter matrix. However, compared with ISS, KPQ prunes non-distinctive points using the ratio between the maximum lengths along the first two principal axes. There are several well-known *adaptive-scale* detectors such as Laplace Beltrami Scale Space (LBSS) and MeshDoG. MeshDoG[147], proposed by Unnikrishnan and Hebert, applied the Different-of-Gaussian operator to 3D mesh to build scale space and MeshDoG is capable of extracting keypoints under high curvature surfaces. For more detailed comparative analysis of existing 3D keypoint detectors, readers are recommended to read the survey paper from Tombari et al. [143].

  Back in 1997, Johnson and Hebert [73] proposed Spin Image which is a data level descriptor that is used to match surfaces represented as surface meshes. The Spin image has been successfully employed in 3D object recognition systems under cluttered environments. After almost one decade, since 2009, with the development of low-cost consumer-level RGB-D sensors, geometry information of the environment can be easily captured thus the importance of 3D shape descriptors have gained recognition again. Various 3D descriptors have been proposed in the literature and are currently available in the Point Cloud Library (PCL) [128]. The 3D geometric descriptors are often divided into two categories: local descriptors which describe the local geometric information in the neighbourhood of the detected keypoints and global descriptors which represent the geometric knowledge of a whole object. Typical local descriptors include Persistent Feature Histogram (PFH)[129] and Normal Aligned Radical Feature (NARF)[16]. Viewpoint Feature Histogram(VFH)[130] and Clustered VFH (CVFH)[4] are some of the popular global descriptors. Since a global descriptor is not suitable to recognise objects under cluttered scenarios, in this study the focus is on local geometric descriptors. Rusu et al.[129] proposed PFH which is a multi-dimension histogram which characterises the local geometry of a given keypoint p in its local neighbourhood region. PFH is invariant to position, orientation and also the point cloud density. Based on PFH, Rusu et al.[127] proposed FPFH and which improves the complexity of PFH, from $O(k^k)$, to $O(k)$ where k is the number of points in the neighbourhood of a given keypoint. Signature of Histograms of OrienTations (SHOT) descriptor proposed by Tombari et al.[144] is another example of a widely used local surface descriptor. SHOT encodes the histograms of the surface normals in different partitions in the support region and the dimension of the SHOT is equal to the product of the division in radial, azimuth and elevation dimension. Guo et al. provide a thorough survey about 3D descriptors and their performances in [60].

- **RGB-D feature**

  As information from both the RGB/grayscale camera and depth camera are complementary, it is possible to combine appearance and geometric information to build descriptors and further improve the matching performance. Lai et al. [85] proved that by joining RGB and depth channels together using spin image[73] and SIFT[96], better recognition performance can be achieved for object recognition when compared with using a single channel only. Tombari et al. [142] introduced colour information to SHOT[144] to develop C-SHOT(Color-SHOT) with the aim of improving the feature matching accuracy performance using appearance information. Nascimento et al. [107] proposed a binary RGB-D descriptor BRAND (Binary Robust Appearance and Normals Descriptor) which encodes local information as a binary string thus making it possible to achieve better performance and low memory consumption. They have also demonstrated the rotation and scale invariance capability of BRAND. More recently, Feng et al. [51] proposed LOIND which encodes the texture and depth information into one descriptor supported by orders of intensities and angles between normal vectors, in addition to the spatial sub-divisions.

Feature Matching and Indexing After extracting the features from object models and input images(2D or 3D), a straightforward solution to identify the correspondences is to simply perform a brute-force search of all descriptors. Unfortunately, this nearest neighbour searching strategy is unrealistic considering the computational complexity. Thus researchers have to use more efficient data structures such as KD-tree or hashing to improve the efficiency of searching. This subsection reviews two types of nearest neighbour searching algorithms: tree-based algorithm such as Approximate Nearest Neighbour (ANN)[102] and hashing-based algorithm such as Locality Sensitive Hashing (LSH)[55]. Feature indexing using visual vocabulary is also discussed briefly.

- **Tree structure based algorithms**

  KD-tree, proposed by Friedman et al..[54], is a binary tree which stores a database of k-dimensional points in its leaf nodes. KD-tree uses spatial partitions and recursive hyperplane decomposition to provide an efficient way to search low-dimensional data *exactly*. Compared with brute-force approach, while still guaranteeing that the nearest neighbour can be found, KD-tree only requires $O\left(N \log N\right)$ computational complexity to construct the tree for N points and $O\left(N^{1-\frac{1}{k}}\right)$ for querying an input point. For high-dimensional feature descriptor, using original KD-tree may end up visiting a large number of additional branches thus degrading the performance. [10] proposed a variant of KD-trees together with a priority queue which relaxes the search requirement to allow the return of *approximate* nearest neighbours. Another idea is to generate multiple randomised KD-trees and process the query in all trees using a single priority queue across them. Muja and Lowe[103] attempted to automatically select algorithm configurations and parameters for desired performances using cross-validation.

- **Hashing based algorithms**

  Hashing algorithms are another type of efficient nearest neighbour search algorithms.

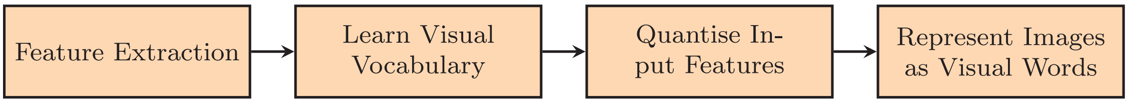


FIGURE 2.4: Typical flowchart of obtaining visual vocabulary representation.

Motivated by the inadequacy of existing exact nearest-neighbor techniques to provide sub-linear time search results for high-dimensional data, randomised approximate hashing-based similarity search algorithms have been explored. The idea in approximate similarity search is to trade off some precision during the search to reduce the query time. Locality Sensitive Hashing (LSH), proposed by Indyk and Motwani [71], offers sub-linear time search by hashing highly similar examples together in a hash table. The idea is that if one can guarantee that a randomised hash function will map two inputs to the same bucket with high probability only if they are similar, then given a new query, one needs to search only the colliding database examples to find those that are most probable to lie in the input's near neighbourhood.

- **Visual vocabulary**

  Unlike the above approaches where the individual feature is queried w.r.t the database, visual vocabulary provides an alternative way to identify the similarity between two *images* instead of two features. In fact, before the emergence of deep learning algorithms in recent years, BoW based approaches were among state-of-the-art methods for object recognition problems. Research on BoW was quite active in some of the critical issues such as feature encoding and visual word learning[162, 161, 36, 90]. This subsection provides the standard pipeline of Visual BoW based recognition approach and discuss some of the most influential work.

  Fig. 2.4 presents the pipeline of standard visual vocabulary algorithms. Instead of using the extracted features directly, there is a need to conduct three additional steps: 1) learning/building the visual vocabulary(dictionary) from the dataset, 2) quantising the extracted descriptors using the vocabulary and 3) encoding a new descriptor of the whole image instead of the individual feature. K-means clustering is a popularly used method to perform the visual vocabulary learning. After initialising the k cluster centres with randomly selected features in the corpus, the algorithm iterates updating each point's cluster until it converges. The feature quantization research began with hard quantization which was later developed into soft quantization and then to sparse coding. Sparse coding based methods have been proven to provide better results for object recognition. Wang et al. presented Locality-constrained Linear Coding (LLC)[155] which applies locality constraints to select similar basis of local image descriptors from a dictionary, and learns a linear combination weight of these bases to reconstruct each descriptor. LLC is easy to compute and it demonstrates superior image classification performance. Sivic and Zisserman[134] proposed quantizing local image descriptors for the sake of rapidly indexing video frames with an inverted file. They showed the potential benefits of exploiting a Term Frequency-Inverse Document Frequency (TF-IDF) weighting on the words, which de-emphasises those words that are common to many images and implementing a stop-list which ignores extremely frequent words that appear in nearly every image.

Hypothesis Verification from Geometric Information Once the correspondences have been identified, it is essential to verify whether those matches satisfy a consistent geometric configuration. This means that the locations and scales of corresponding features in both the query image and the template image(or object model) are related by a fixed geometric transformation which can be translation transformation, rotation transformation, affine transformation and similarity transformation. In practical cases, *homography transformation*, a projection of a plane onto another plane, is frequently used. The methodology of using homography transformation in order to locate the consistent correspondences is briefed as below. Homography transformation from a point $\mathbf{p}_a$ to the associated point $\mathbf{p}_b$ can be represented as follows:

$$\mathbf{p}_b = \frac{1}{z'_b}\mathbf{x}'_b \text{ with } \mathbf{x}'_b = \mathbf{H}\mathbf{x}_a$$
$$\begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix} = \frac{1}{z'_b}\begin{bmatrix} x'_b \\ y'_b \\ z'_b \end{bmatrix} \qquad \begin{bmatrix} x'_b \\ y'_b \\ z'_b \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}\begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix} \tag{2.1}$$

Given a set of matches, Direct Linear Transform (DLT) can be used to compute the homography matrix, and the detailed algorithm is presented in Chapter 4.1 of Hartley and Zisserman's book titled *Multiple View Geometry in Computer Vision*[63]. Given correct correspondences, it may sound trivial to identify the set of features which follow a consistent geometric transformation, however; in reality, it is non-trivial to remove the outliers in a set of feature correspondences. This section reviews two fundamental techniques which are popularly used in removing outliers in the raw data: RANdom SAmple Consensus(RANSAC) [52] and General Hough Transform(GHT)[15].

RANSAC is a non-deterministic algorithm that operates in a hypothesize-and-test framework. Thus, it only returns a "good" result with a certain probability, but this probability increases with the number of iterations. The procedures of RANSAC is presented in Fig. 2.5.

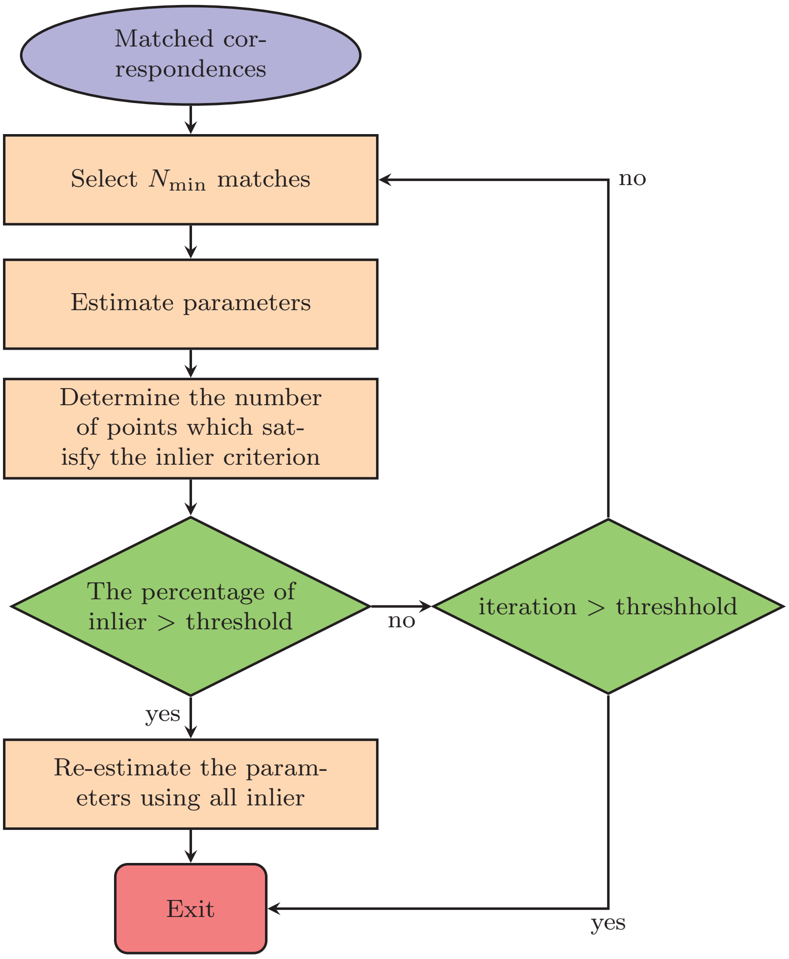

Matched correspondences
Select $N_{\min}$ matches
no
Estimate parameters
Determine the number of points which satisfy the inlier criterion
The percentage of inlier > threshold
no
iteration > threshhold
yes
Re-estimate the parameters using all inlier
Exit
yes

FIGURE 2.5: Flowchart of RANSAC to remove outliers in raw features matches.

Hough Transform was originally introduced in 1962 as an efficient method for finding straight lines in images[151]. Ballard, later on, showed how to generalise this idea to detect arbitrary shapes, leading to the GHT[15]. The basic idea of this extension is that the observed single feature correspondences vote for the parameters of the transformation would project the object in the model image to the correct view in the test image.

2.1.2 Category-Level Object Recognition

First, some standard paradigms for *image classification* problems are introduced. As Yann LeCun and Marc' Aurelio Ranzato summarised in [88], the traditional image classification pipeline is presented in Fig. 2.6. SIFT or HoG features are used to extract low-level information of the object and the clustering methods such as K-mean is adopted to learn more abstract level representation. Support Vector Machine (SVM)[150] is a very popular choice for the classifier step.

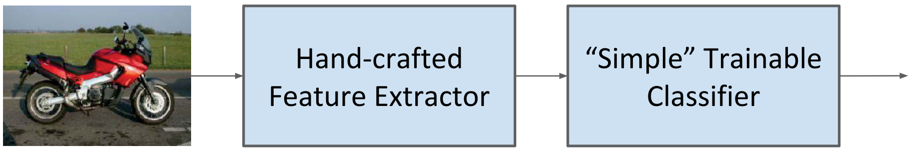


FIGURE 2.6: Traditional image classification pipeline.

After 2011, deep learning became the de-facto technique in image classification and recognition. Instead of using the hand-crafted features, researchers tend to learn the feature representation automatically from neural networks such as CNN and Residual Neural Network (RNN). Fig. 2.7 presents comparable framework for the deep learning based pipeline. In this framework, the common hierarchical feature structure is constructed from pixel, edge to texton, motif, part and object in a lower-to-higher order. Please refer to [83] and [21] for detailed network structure.

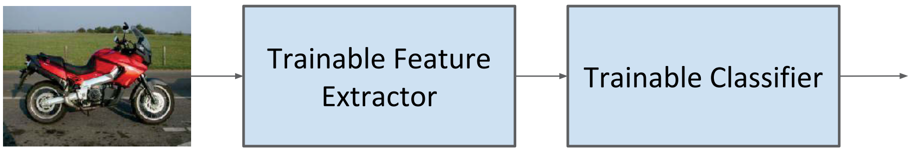


FIGURE 2.7: Deep learning based image classification pipeline.

However, in order to detect/recognise objects in the image, before inputting image patches which may contain the targeting object into the pre-trained classifier, firstly, the Region-of-Interest (ROI) needs to be identified from the input image. Sliding window approach is a popular technique where windows of different shapes, sizes are scanned on an image in different scales to detect the object. Visual saliency[25] and object proposal[68] are two favourite tools to speed-up the sliding window detection.

2.1.3 6D Pose Estimation

There is a considerable body of literature in the area of 6D pose estimation for object detection, including instance and category recognition, rigid and deformable objects, and coarse (quantized) and accurate (6D) poses. This subsection reviews some of the feature-based 6D pose estimation approaches which are more related to the work presented in this thesis. For template based approaches, interested readers are referred to [136, 69, 65].

Perspective-n-Point Problem Perspective-n-Point (PnP) is the problem of estimating the pose of a calibrated camera given a set of n 3D points in the world and their corresponding 2D projections in the image. The camera pose consists of 6 degrees-of-freedom (DOF) which are made up of the rotation (roll, pitch, and yaw) and 3D translation of the camera w.r.t the world. Estimating the relative pose of the object w.r.t the sensor is an equivalent problem to the PnP problem given the 3D object model and object coordinate frame. The solutions for the PnP problem are classified into 1) iterative methods and 2) noniterative methods. Noniterative methods are efficient however more unstable in the presence of noise data. Pose from Orthography and Scaling with ITeration (POSIT)[37], proposed by Dementhon and Davis, is one of the most popular iterative PnP solution. In canonical perspective, the projection equation can be written as below where $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector, $[xyz1]^\intercal$ is the homogeneous 3D coordinate of the feature and $[uvw]^\intercal$ is the coordinate of projected feature.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.2}$$

In POSIT, an initial value is assigned on w prior to computing $\mathbf{R}$ and $\mathbf{t}$. There are at least 4 pairs of non-coplanar points used to compute $\mathbf{R}$ and $\mathbf{t}$. Afterwards, w is updated using the already computed $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$. These steps are iterated until the convergence. [7][1] proposed non-iterative methods which can determine 6DoF pose with fewer than 5 points. However the solution tend to be unstable in practice due to the lack of redundant information. The stability of the non-iterative methods can be improved by incorporating redundant points. The well-known DLT algorithm[63] achieves relatively accurate results from a large number of points. However, as pointed out by Lepetit et al. [91], many non-iterative methods are slow due to high computational complexity for processing large point sets . Lepetit et al. [91] also presented an efficient non-iterative algorithm with linear complexity in n by expressing the solution as the weighted sum of null eigenvectors. Li et al. presented a robust non-iterative solution of the PnP problem(rPnP). rPnP works well for both non-redundant point sets ($n < 5$) and redundant point sets. rPnP achieves even better results compared with DLT, ePnP and ePnP with Gauss-Newton, and its computational complexity grows linear with n.

Pose Estimation from 3D-3D Points Set When using an RGB-D sensor, in addition to the 2D coordinate on image space ($[u, v]$), the 3D coordinate of the feature can be acquired using the depth information. Therefore, the relative pose estimation problem becomes trivial given 3D-3D correspondences. Eggert et al.[44] review 4 popular and efficient solutions for this problem. The first solution was developed by Arun et al. [9] which is based on computing the singular value decomposition (SVD) of a matrix derived from the standard $[\mathbf{R}, \mathbf{t}]$ representation. This approach is adopted in our work presented in chapter 3. Horn et al. [67] presented a similar approach in which the orthonormal properties of the rotation matrix were exploited to compute the eigensystem of a different derived matrix. The third algorithm, also developed by Horn[66], involves computing the eigensystem of a matrix related to representing the rotational component as the quaternion. Walker et al,[154] presented another approach where the eigensystem is analysed with the translation and rotation components represented using dual quaternions. All of these four methods can compute the relative transformation in closed-form.

The rest of the text in this subsection is dedicated to discussing some representative systematic object detection and pose estimation works in this field, especially the ones which are more related to robotic research.

Drost et al. [43] introduced an efficient, stable and accurate method to find 3D free-form objects in point clouds. Different to local feature based approaches as was discussed in section 2.1.1, Drost et al. created a global model based on oriented point-pair features and the model is matched locally using a fast voting scheme. The global representation leads the model to be independent from local surface information. Thus very fast matching results can be achieved due to locally reduced search space. The proposed method was evaluated on both synthetic data and real data, and they demonstrated that with a slight or even no sacrifice of the recognition performance, the proposed method decreases the time consumption significantly.

Collet et al. [32] presented an object detection and pose estimation framework for object manipulation task, named Multiple Object Pose Estimation and Detection (MOPED). From a theoretical perspective, the key contribution of this work lies in Iterative Clustering Estimation (ICE) which is a novel algorithm that iteratively combines feature clustering with pose estimation. ICE handles the common outlier in feature matching and clustering quite well by considering pose estimation simultaneously. On practical perspective, MOPED is a fully optimised, robust and scalable framework where GPU and CPU are hybrid and parallelism is exploited at all level. They provided extensive experimental results demonstrating state-of-the-art performance regarding recognition, scalability, and latency in real-world robotic applications.

Hinterstoisser et al. [65] presented a template based approach for less-textured object detection under a cluttered environment which does not require a complicated training stage. The core contribution of this work is a novel image representation method for template matching which is designed to be robust to small image transformations using spread image gradient. Complementing on the additional depth information provided by the sensor, this framework demonstrates even better performance by considering surface normal information. Their work is also fully optimised using the modern parallel SSE instruction and GPU computation. Their approach outperforms state-of-the-art methods on recognition rate and speed, especially for less-textured objects in heavily cluttered environments .

Aldoma et al. [5] presented a tutorial on 3D object recognition and 6D pose estimation using the capabilities present in the Point Cloud Library. The pipeline is very similar to what was discussed in Section 2.1.1 with an additional global pipeline. They adopted uniform sampling to extract the keypoints in the point cloud and various descriptors available in PCL. Simultaneously, global descriptors such as CVFH(Clustered Viewpoint Feature Histogram) are used to describe the keypoints and match them against the features on the models. The relative pose is estimated using ICP.

Tang et al. [138] presented a fused textured object recognition and pose estimation pipeline by fully utilising both RGB and depth information provided by a Kinect sensor. During the training phase, a sparse feature point cloud is built as the object model similar to the work presented in [32]. Additionally, a global colour model for each object is trained using hue histogram. In the detecting phase, the support plane where the objects are placed on is extracted and removed first. The remaining point cloud is clustered into multiple groups and a colour descriptor is extracted from each cluster. Meanwhile, point features from each cluster are extracted and matched against the trained model. In the end, the pose is estimated using a Levenberg-Marquardt nonlinear least square algorithm.

2.2 Active Object Detection and Next-Best-View Selection

Previous sections of this chapter presents the recent advancements in object recognition and pose estimation from a single view observation (i.e. an image). However, in real life, a single view of the environment may not contain sufficient information to recognise target objects unambiguously. With a sensor (i.e. monocular or RGB-D camera) mounted on a mobile robot platform, it is possible to move the sensor actively in the environment to capture more information from different viewpoints and hence to potentially increase the confidence of the detected objects and their 6DoF relative pose estimates. The text in this section is organised into two parts depending on whether the motion and observation uncertainties are considered while moving the sensor. If the motion and observation uncertainties are not taken into consideration, this problem is regarded as a Next-Best-View(NBV) problem[125]. When uncertainties are taken into consideration, the same problem is formulated in a *planning under uncertainty* framework using theoretical tools such as Partially Observable Markov Decision Process(POMDP) [140, 74]. Planning with uncertainties cases can be further classified into two different type of approaches: 1) sampling-based methods[75, 28] and 2) optimisation-based methods[148, 70].

2.2.1 Next-Best-View for Active Object Recognition

Next-Best-View problem is widely used in surface reconstruction[114], object modelling[53, 81] and scene exploration[157, 98, 23]. The term *active vision* is frequently used in this subsection which covers not only the active object recognition but also tasks such as active inspection and active search.

In[125], Roy et al. categorised active object recognition systems based on 1) representation schemes and 2) planning strategies. There are two popular representation schemes:

1. view-based representations which encode information about different viewpoints (observations) of a 3D object;

2. part-based representation which considers an object in terms of its separated parts;

The NBV planning strategy is also classified into two categories:

1. systems which take the NBV to minimise an ambiguity function;

2. systems which incorporate explicit planning algorithms

They also performed a comparative analysis of the active object recognition systems using different criteria such as feature types for modelling and recognition, speed and efficiency, and object model representation. The work presented Borotschnig et al. [26] uses an appearance-based information in an online fashion. Their work shares the similar objective as the work presented in this thesis. They use a parametric eigenspace and augment it with probability distribution to capture possible variations in input images.

Chen et al.[30] presented a broad survey of developments in active vision in robotic applications. Their paper summarises progress of various applications such as object recognition and modelling, site reconstruction and inspection, surveillance, tracking and search, as well as robotic manipulation and assembly, localisation and mapping, navigation and exploration. Denzler and Brown[38] presented remarkable work in selecting sensor data for active object recognition and state estimation with the objective of reducing uncertainty in the state estimation process, rather than an estimator-specific metric. This makes their approach more general and independent from the existing state estimator. Their work also proposes a method for selection of optimal sensor parameters for iterative state estimation in static systems. In [38], the capability of the proposed approach was demonstrated using a state estimation problem of a static system using active gaze control. More importantly, contributions in [38] are not limited to only active object recognition.

In [11], Atanasov et al. address the problem of object classification and pose estimation of semantically important objects by actively controlling the viewpoint of a mobile depth sensor. The proposed a novel static detector, Viewpoint-Pose Tree(VP-Tree), combines object detection and pose estimation in 3D environment. In order to reach the goal of non-myopic view planning, this problem is formulated as a POMDP problem and a point based approximation algorithm proposed in [61] is employed to solve it. The proposed approach is verified in both simulation and real world experiments with an RGB-D camera mounted on a PR2 robot. The comparative results against greedy viewpoint selection and single-viewpoint detection show the superior performance of their work.

Potthast and Gaurav proposed two important information-theoretic based approaches for active path planning. In [117], they proposed an information gain-based variant of the NBV problem to recognise objects under occlusions. Their method utilises a belief model of the unobserved space to estimate the expected information gain of each possible viewpoint which provides a more precise estimation of the future visibility of occluded space. Furthermore, more accurate prediction of the potential information gain of new viewing positions can be incorporated into the planning step. Under extremely cluttered scenarios, the proposed approach can reduce the number of unobserved cells faster than traditional approaches. More recently, Potthast et al. [118] proposed an information-theoretic framework which combines and unifies two common techniques: online feature selection for reducing computational costs and view planning for resolving ambiguities and occlusions. This method adaptively selects one strategy over the other either by selecting the features that are most discriminative or by moving the sensor to a new viewpoint that optimally reduces the uncertainty of recognition outcomes. This two-step process allows keeping the overall computation cost minimal while simultaneously increasing the recognition accuracy. Extensive empirical studies on a large RGB-D dataset, and with two different sets of features, have validated the effectiveness of the proposed framework. Capitalising on the strengths of deep learning, Doumanoglou et al.[42] presented a complete framework for both single shot-based 6D object pose estimation and NBV prediction based on Hough Forests. Instead of using manually designed local features, they propose to use unsupervised features that can be learnt from depth-invariant patches using a Sparse Autoencoder and a 6D Hough voting scheme for pose estimation. The active vision strategy is built on Active Hough Forests for estimating the NBV.

2.2.2 Planning under Uncertainty

Decision making under uncertainty is a crucial requirement for most robotic systems. The uncertainty in robotic systems usually stems from a) motion uncertainty which affects the system dynamics and b) observation uncertainty caused by the noise in the sensor measurement. In this case, a state estimation module is used to provide the probability distribution over possible states, also known as *belief space*[2]. This subsection presents literature on active object detection using general belief space together with relevant literature which focus more on visual perception[35, 13, 109].

[2] This terminology is often referred as *information space* in [35, 2] interchangeably

The existing approaches towards planning under uncertainty are categorised into two different types: (a) sampling-based methods[28, 75] (b) optimisation based methods[149, 115]. Some well-known sampling-based methods are firstly reviewedsuch as Rapid-exploring Random Belief Tree(RRBT)[28] and Feedback-based Information Roadmap(FIRM)[2]. Latter part of this subsection focuses on the optimisation-based algorithms such as Linear Quadratic Gaussian based methods [148, 149] and Sequential Quadratic Programming (SQP) formulated solutions. This section also discusses some of the visual perception related planning work such as [35] and [109].

Sampling-based Methods Rapid-exploring Random Tree (RRT)[87] and Probabilistic Roadmap (PRM)[76] are the most influential sampling-based motion planning algorithms where uncertainties are ignored. Even though the idea of connecting points sampled randomly from the state space is similar in both approaches, these two algorithms differ in the way they construct a graph connecting these points. Based on RRT, Karaman and Frazolli[75] proposed Rapid-exploring Random Graph (RRG) as an extension. Besides the nearest connections, in RRG, the new samples are also connected to every node within a sphere thus it is locally refined with each added sample. This refinement guarantees that the RRG algorithm will contain all possible paths through the environment within the limit of infinite samples. RRT*, a variant of RRG, is also proposed by Karaman and Frazolli[75] which guarantees to converge to the optimal path by only keeping the edges in the graph that result in lower cost at the vertices in the sphere. However, all of the above approaches assume the fully deterministic dynamics and therefore, are not suitable for planning with stochastic properties.

In order to address the state uncertainty, Roy proposed two variants, Belief Roadmap (BRM)[119] and RRBT[28], with Prentice and Bry respectively. Targeting at partial observability issues, BRM, proposed by Prentice and Roy, simulates measurements along candidate paths and chooses the path with minimal uncertainty. However, BRM still assumes the mean of the system is controllable which means that while the path is being executed, the controller is always capable of driving the state estimate back to the desired path and this can be an unattainable goal for a practical robot and environment. RRBT, proposed by Bry and Roy, is an important work in motion planning under uncertainty. Given an intractable problem involving non-trivial dynamics, spatially varying measurement properties and obstacle constraints, Bry and Roy simplify the motion plan to a nominal trajectory stabilised with a linear estimator and controller, Linear Quadratic Gaussian (LQG) controller, which thus allows prediction of the belief of future states given a candidate nominal trajectory. Both sampling-based algorithms and linear control and estimation schemes have been shown to scale well with dimensionality so that RRBT can be extended easily to more complicated systems. Agha-mohammadi et al. presented Feedback-based Information Roadmap (FIRM), a multi-query approach for planning under uncertainty which is a belief space variant of probabilistic roadmap methods. Exploiting feedback controllers, they reduced it to a tractable FIRM Markov Decision Processes (MDP) problem that can be solved using standard Dynamic Programming (DP) techniques. FIRM utilises feedback controllers to create reachable node regions in belief space. An important consequence is that FIRM preserves the optimal substructure property on the roadmap and thus overcomes the curse of history in the original POMDP problem. They showed an instantiation of the abstract FIRM framework using Stationary Linear Quadratic Gaussian (SLQG) controllers and illustrated the construction and planning results on it.

Optimisation based Methods Optimisation techniques can be used to compute a robot trajectory that is optimal under some specific metrics (e.g., smoothness or length) and at the same time satisfies various constraints, e.g., collision-free and dynamics constraints. Some algorithms assume that a collision-free trajectory is given and it can be refined or smoothened using optimisation techniques. Van den Burg et al.[148] proposed Linear Quadratic Gaussian Motion Planning (LQG-MP) which is one of the most popular optimisation based planning algorithms under uncertainties. By representing the beliefs as Gaussian distributions and approximating the belief dynamics using an Extended Kalman Filter (EKF) and more importantly by formulating the objective function as a quadratic function in belief space, they are able to characterise the probability distribution of the state of the robot along the path. In this work[148], Van den Burg et al. studied the performance of LQG-MP with simulation experiments under different scenarios and Dijkstra's algorithm based LQG-MP precomputed roadmaps were used to find optimal paths efficiently.

However, in [148], even though the *maximum likelihood observation* assumption is relaxed, LQG-MP is only capable of evaluating the probability of success of a given trajectory rather than constructing an optimal one. To overcome this limitation, Van den Berg et al.[149] proposed a new approach to motion planning considering both motion and observation uncertainties by computing locally optimal solution towards a continuous POMDP problem. Similar to [148], the beliefs are represented as Gaussian distributions. Non-linear motion model and observation model are formulated by EKF, using iterative LQG (iLQG) proposed by Todorov and Li [141], the proposed approach iterates with second-order convergence towards a linear control policy over the belief space that is locally optimal w.r.t the cost function. This approach is demonstrated and verified in simulation for holonomic and non-holonomic robots manoeuvring through environments with obstacles. The main drawback of this algorithm is the associated computational complexity of $O(n^6)$

In order to reduce the complexity of the algorithm presented in [149], Patil et al. proposed a novel covariance-free trajectory planning algorithm with complexity $O(n^3k)$ where n is the dimension of the space and k is the number of steps in the planned trajectory. Instead of using both mean and covariance to describe the uncertainty of the belief, in [110], the locally optimal trajectory is computed without including the covariance. Therefore, the dimensionality of this problem scales linearly in the state dimension instead of quadratically. The main reason for ignoring covariance is because of the recent progress in numerical optimal control such as automatic differentiation[59] and modern convex solvers[41]. This algorithm is validated on different applications, for example, manipulator planning, parameter estimation for dynamical systems and active SLAM. This approach is $400\times$ faster than the traditional trajectory based optimisation methods.

In order to avoid the *maximum likelihood observation* assumption, Indelman et al. presented a probabilistic framework where the motion planning is computed from the *Generalised Belief Space* (GBS) which contains both robot space and landmarks. Their work consists of two layers: 1) an inner layer which estimate the belief given a set of control input and 2) an outer layer which optimises the control input by minimising the designed objective function. The decision-making process is entrusted to a Model Predictive Control (MPC) framework. More importantly, they assume the future predicted observations as binary random variables. Under a specific objective function formulation, through non-trivial derivation, these binary random variables are later eliminated in the optimisation step. The proposed approach is applied in uncertainty-constrained exploration tasks where the vehicle is manoeuvred in an unknown environment.

The above methods have been successfully applied to multiple robotic applications, e.g., BRM has been used for UAV navigation under a GPS-denied environment[64, 12]. Based on RRT*, Costante et al. proposed an online perception-aware path planning framework which considers both geometric information and photometric information of the environment. In order to take the texture information into account, they adopt dense, direct SLAM[79] to compute the photometric information gain from the intensity values of every pixel in the image. The effectiveness of the proposed approach is demonstrated on a quadrotor platform with applications such as vision-based localisation, dense 3D reconstruction and online perception-aware planning. Pathak et al. also raise another important issue, the data association in reasoning feature observation, in their recent work[109]. While previous approaches assume that data association is perfect, they proposed Data Association awared Belief Space Planning (DA-BSP) which explicitly considers the uncertainty of DA in belief propagation. In particular, they show that due to perceptual aliasing, the posterior belief becomes a mixture of probability distribution functions. They design a cost function which measures the expected level of ambiguity and posterior uncertainty. Therefore, this algorithm is applicable to robust active perception and autonomous navigation in a perceptual aliased environment.

2.3 Summary

This chapter reviews the progress in both single-view and active object detection and pose estimation problems. The general pipelines for instance-level object detection, category-level object detection and 6 DoFs pose estimation are summarised. Because of the objective and emphasis of this thesis, this chapter only illustrates some representative work in instance-level object recognition and pose estimation in detail. Based on whether the motion and observation uncertainties are considered, the active object detection and pose estimation approaches is separated into two categories: NBV for object recognition and active object recognition in belief space. Corresponding to the contributions in subsequent Chapter. 6 and Chapter. 7, influential work in both NBV and planning under uncertainty are reviewed and the key contributions are discussed.

Part I

Single-View Object Detection and Pose Estimation

Chapter 3

Textured Object Recognition and Pose Estimation

As was discussed in Section. 2.1.1, given a sufficient number of reliable correspondences, using a monocular camera, target objects can be recognised and localised even in slightly occluded environments[31]. However, issues such as robustness and efficiency still remain open. Using modern day consumer-level RGB-D sensors, geometric information can be acquired to provide extra information to address these issues in different ways.

The goal of this chapter is to develop an object recognition and pose estimation framework for textured objects using an RGB-D sensor. By taking advantages of the 3D information from the RGB-D sensor, a modular, robust and efficient perception system is presented which demonstrates reliable performances under cluttered environments. The depth information helps improve the consistency of the feature clustering results; it speeds-up pose estimation while holding the accuracy. Instead of using RANSAC[52], a novel outlier rejection algorithm is proposed by formulating a relative graph from the correspondences which also significantly reduces the computational time.

This chapter is structured as follows: Section. 3.1 formulates the input, output of the system and defines the variables and terminologies which will be used later; Section. 3.2 illustrates the pipeline of the proposed framework and highlights the key innovations of this work; Section. 3.3 identifies the improvements of the proposed methods and also presents the object detection results using this framework. Section. 3.4 concludes this chapter and discusses the limitation of this work.

3.1 Problem Formulation

The input information from an RGB-D sensor consists of a colour image I_{colour} and a depth image I_{depth}. The organised dense point cloud P can be readily computed using I_{colour}, I_{depth} and the intrinsic calibration parameters. Before the recognition procedures, the model for each object is built using off-the-shelf Structure-from-Motion toolboxes (*bundler* [135] and VisualSFM[158]), in a manner similar to MOPED [32]. Each model M consists of a set of 3D features f_i and each f_i includes the 3D coordinate $p_i = [x_i, y_i, z_i]$ w.r.t object coordinate frame and an associated SIFT descriptor $\mathbf{d}_i \in \mathbb{R}^{128}$. The output of the proposed algorithm is the object recognition hypothesis H which consists object identity O and its relative pose $[\mathbf{R}, \mathbf{t}]$ w.r.t the camera coordinate frame. f^M with superscript M is used to denote features from pre-trained model and f^O to denote features captured from observation. c_{ij}^{mn} represents the correspondence between feature n in model M_m and feature j in cluster i.

3.2 Algorithm Pipeline

This section illustrates the proposed framework and the methodology in each step. The highlighted red block in Fig.3.1, represents the outlier rejection method, the key contribution of this chapter 3.2.4.

3.2.1 Support Plane Subtraction

In our experiments, all target objects are assumed to be placed on one support plane. Given the raw input point cloud P, the largest plane P_π is extracted from the scene as the support plane using RANSAC[52] model fitting. Unlike the experimental set-up in Willow dataset[1] where a chessboard marker is placed on the table as an assistance for plane extraction, as shown in Fig.3.2, our assumption is more general, and this is also a realistic assumption which is acknowledged in object recognition and pose estimation work such as Narayanan's recent work[106]. From practical implementation viewpoint, another issue is that the chessboard based plane detection method is not as robust as the RANSAC-based method. As shown in Fig. 3.3, the corner points can be extracted correctly from the right-hand image, Fig. 3.3(a), which contains a larger chessboard, however, lots of difficulties are encountered when trying to extract corner points from Fig. 3.3(b).

[1] https://www.willowgarage.com/blog/2011/02/28/nist-and-willow-garage-\ solutions-perception-challenge

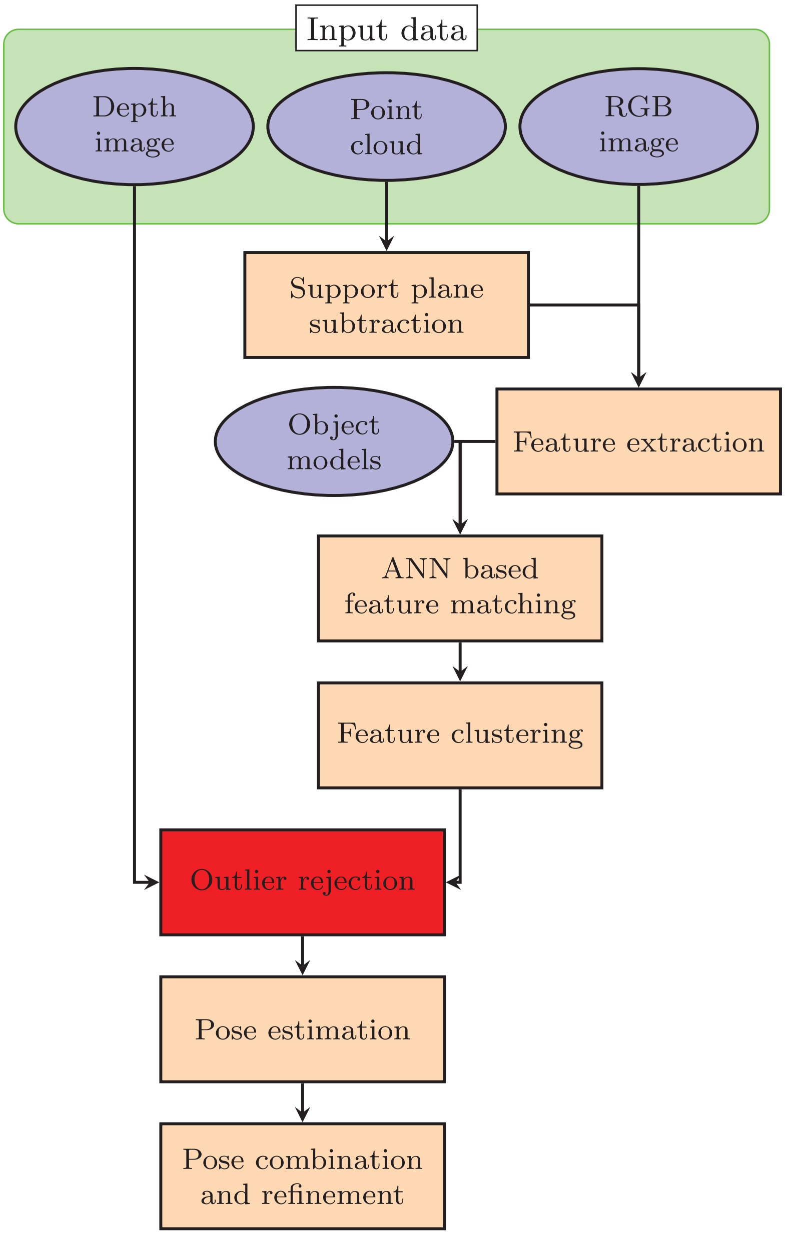
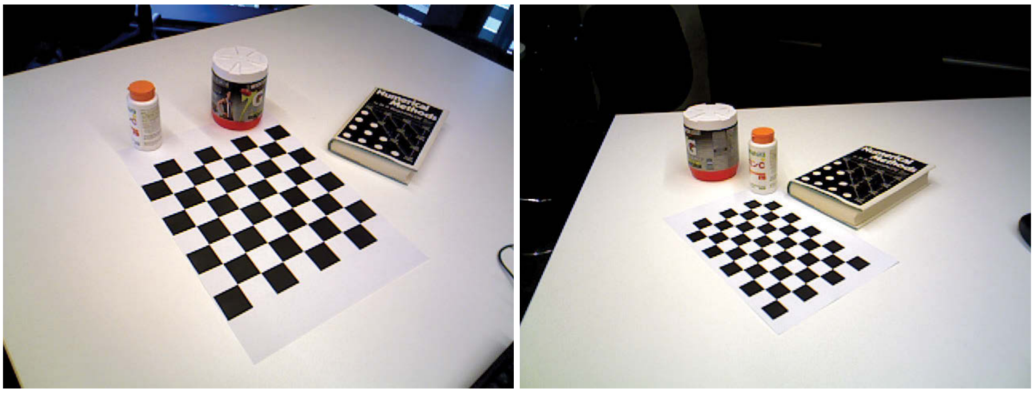


FIGURE 3.1: Pipeline for single view textured object detection and pose estimation

FIGURE 3.2: Sample image from Willow dataset

Besides, RANSAC-based plane extraction method shows similar computational time compared with the chessboard based method.



(a) An easy case where corners can be extracted robustly.

(b) A difficult case where corners can not be extracted robustly.

FIGURE 3.3: Example images from reconstructed scenarios similar as Fig. 3.2.

The normal vector $\mathbf{v}_\pi$ of extracted plane P_π is computed further and the centre point p_π of P_π is selected as the original point. For all other 3D point p_i, the projected vector of $\mathbf{v}_i = p_i - p_\pi$ onto normal vector $\mathbf{v}_\pi$ is computed, denoted as $\hat{\mathbf{v}}_i$ and

$$\hat{\mathbf{v}}_i = |\mathbf{v}_i|(\mathbf{v}_i \cdot \mathbf{v}_\pi)\mathbf{v}_\pi \tag{3.1}$$

The point p_i is regarded as above the plane if $\hat{\mathbf{v}}_i > \mathbf{v}_\tau$. In our experiments, $\mathbf{v}_\tau$ is set to be positive and $|\mathbf{v}_\tau| = 0.005(m)$. A binary mask image I_{mask} is generated where only points above the plane with available depth information are assigned with 1 and vice versa.

3.2.2 Feature Extraction and Matching

The widely used SIFT[96] feature was adopted and features are extracted using the generated mask image. Provided with both RGB and depth information, the feature coordinate in image frame $p_k^I = [u_k, v_k]$ and corresponding 3D coordinate $p_k^C = [x_k^C, y_k^C, z_k^C]$ in camera coordinate system are available and kept for further 3D-3D pose estimation.

In order to reduce the time consumption of finding the correspondences between observation and models, the fast ANN[102] searching method is adopted. Because of the complexity of the environment, outliers are likely to be introduced in this phase while the efficiency is improved. $\mathbf{c}^w$ is used to denotes all the correspondences matched to model w where $\mathbf{c}^w = \cup_{\forall m=w} c_{ij}^{mn}$.

In order to evaluate the matching accuracy, following state-of-the-art feature detector and descriptor algorithms are used to conduct a comparative experiments: 1) SIFT keypoint detector and feature descriptor; 2) SURF keypoint detector and feature descriptor; 3) FAST[123] keypoint detector and SIFT feature descriptor; 4) FAST keypoint detector and BRIEF feature descriptor; 5) ORB feature detector and descriptor. Table. 3.1 tabulates the matching accuracy under each of these methods. Features between the observation and the model are matched using ANN and the outliers are removed using RANSAC with homography constraints. SIFT can obtain comparatively higher number of correct matches given the same number of features compared to all the other methods.

TABLE 3.1: Feature matching accuracy comparison results[2]

Feature	SIFT	SURF	FAST+SIFT	ORB	FAST+BRIEF
Matching Accuracy	0.3287	0.2162	0.1761	0.1913	0.1572

3.2.3 Feature Clustering

For all matches $\mathbf{c}^w$ between observation and model w, geometrically closer features in camera coordinate system are grouped using Mean-Shift algorithm[33]. Different from the 2D space clustering in [32], assuming that the environment is static, 3D distances between each pair of points are fixed from different viewpoints. Therefore, 3D distances based clustering is more robust compared to the clustering in the image space. The comparison results of two clustering methods are

[2] $\text{Accuracy} = \frac{\text{NumberOfCorrectMatches}}{\text{NumberOfExtractedFeatures}}$

presented in Fig. 3.4. 3D distances based clustering (the top row in Fig. 3.4) shows more consistent and reliable results under different ranges. However 2D clustering results (the bottom row in Fig. 3.4) show a comparatively larger number of incorrect clusters and the results tend to be very inconsistent. After clustering, each group G_i consists of a subset of correspondences between the model M and the observation and is hypothesised to contain object instances. Outliers inevitably exist after clustering and therefore they need to be removed during pose estimation.

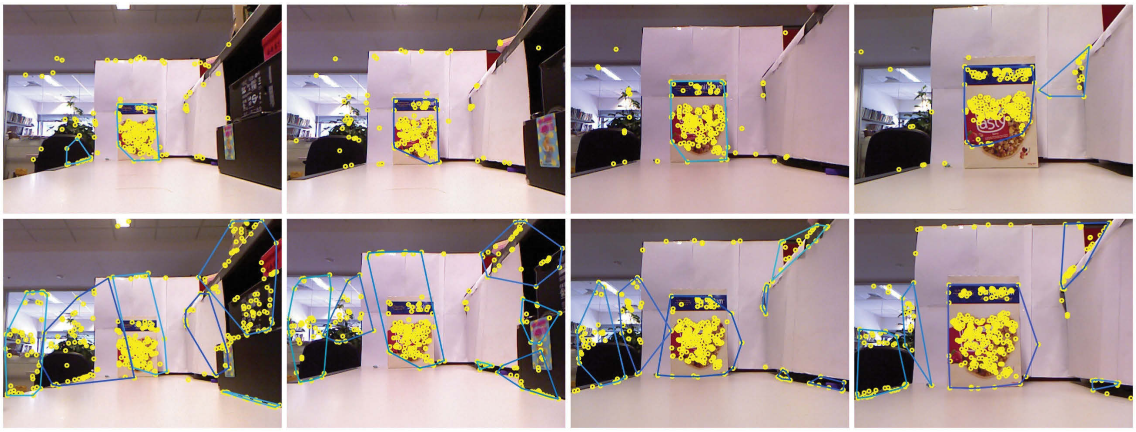


FIGURE 3.4: Comparison results for clustering correspondences for 4 same images under different ranges using two different methods.

3.2.4 Outlier Rejection

Using an RGB-D sensor, it is able to capture 3D information from the scene. Identifying the correct correspondences given a set of 3D-3D matches is a simplified problem compared with [47] and [93] where no initial correspondences are given. The significant differences are: 1) 3D-3D matches are available rather than 3D-2D matches thus the geometrical constraint can be exploited; 2) initial matches are given using SIFT features (note that, whereas in [93] the matches are unknown). This is a typical RANSAC problem which can iteratively estimate relative pose and rejects outliers. Here, a graph-based approach is presented which is able to solve this problem more efficient and is also robust when compared with RANSAC.

Given a set of hypothetical correspondences $\mathbf{c} = \{c_1, c_2, ..., c_n\}$ where c_i consists of 3D coordinate p^M in the model frame and 3D coordinate p^O in the sensor frame. A relation graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is constructed using $\mathbf{c}$ as follows: each vertex v_i denotes a correspondence c_i and the weight of the edge connecting two vertices v_i and v_j is calculated as (3.2):

$$\omega_{i,j} = 1 - \left| \left\| p_i^M - p_j^M \right\|_2 - \left\| p_i^C - p_j^C \right\|_2 \right| \tag{3.2}$$

where $\|\mathbf{p}\|_2$ denotes the L2 norm of vector $\mathbf{p}$. $\omega_{i,j}$ is able to represent the consistency between a pair of correspondences c_i and c_j. Smaller $\omega_{i,j}$ denotes less-consistency and larger $\omega_{i,j}$ means that the 2 correspondences are more consistent w.r.t a unknown rigid transformation. Given this fully-connected weighted undirected relation graph $\mathbf{G}$, in order to find a consistent subset of correspondences, the strongly connected components need to be identified. However, finding an exact solution to this problem is NP-hard. An approximate solution towards this problem is proposed which is fast and capable of generating reliable results under practical environments.

By thresholding on the weight of the each edge, the weighted fully-connected undirected graph $\mathbf{G}$ is converted into a secondary undirected graph $\mathbf{G}$ without weighted edges firstly. If $\omega_{i,j}$ is larger than a pre-defined threshold ω_ε, the two pairs of correspondences are regarded as consistent w.r.t a same unknown transformation $[\mathbf{R}, \mathbf{t}]$ and this edge is preserved. Otherwise it is removed in the graph. All the vertices in this secondary graph $\mathbf{G}$ are traversed to calculate the number of edges n_i^e connected to each v_i. n^e of the most connected vertex v^e in the graph is checked and all the vertices connected with v^e if $n^e > \tau_{\mathbf{n}_v}$ (threshold) are removed. The step is processed iteratively until the termination criteria $n^e < \tau_{\mathbf{n}_v}$ is met and all the remaining vertices are considered as outliers. The algorithm is presented in Algorithm. 1 and vertex v is also used to denote a correspondence c. Compared with RANSAC, this method only needs to compute the number of connected edges for all v_i and a simple algorithm to find the most connected vertices in $\mathbf{G}$. In our experiments, normally, depending on the actual number of consistent cliques in the relational graph, the loop can be finished within less than 5 iterations. The detailed comparison experiments and results are presented in section. 3.3.1.

3.2.5 Pose Estimation

There are several 3D rigid body transformation methods described in [45]. In this work, given 3D-3D correspondences, the least square solution based on SVD [8] is adopted. In order to deal with the noisy depth observations, RANSAC is implemented combined with SVD pose estimation while RANSAC can find the correct correspondences in significantly fewer iterations. SVD based

Algorithm 1: Optimal correspondences searching

```
Input: correspondence set $\mathbf{c} = \{c_1, c_2, ..., c_n\}$ where $c_i = (p_i^{\mathbb{M}}, p_i^{\mathbb{C}})$;
Output: Consistent cliques $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ...\}$ where $\mathcal{C}_i = \{c_{i_1}, c_{i_2}, ..., c_{i_k}\}$
// $\mathbf{ConnectedVertices}(v)$: returns all the vertices linked with $v$;
// $\mathbf{MaxVertex}(\mathbf{G})$: returns the most connected vertex in $\mathbf{G}$;
// $\mathbf{G} = \mathbf{Graph}(\mathbf{c})$: build graph from $\mathbf{c}$;
$v_{max} = \mathbf{MaxVertex}(\mathbf{G})$;
while $|\mathbf{ConnectedVertices}(v_{max})| > \tau_{\mathbf{n}_v}$ do
    $\mathcal{C} = \{v_{max}, v_{connected}\}$;
    $\mathbf{V} = \mathbf{V} - \mathbf{ConnectedVertices}(v_{max})$;
    $\mathbf{G} = \mathbf{Graph}(\mathbf{v})$;
    $v_{max} = \mathbf{MaxVertex}(\mathbf{G})$;
end
```

pose estimation is faster compared with the Levenburg-Marquart optimisation based technique which is used in [31].

3.2.6 Pose Combination and Refinement

There exist two different methods to handle overlapping recognised hypotheses:

1) if multiple adjacent hypotheses H_i have the same recognised object and overlapping poses, it is highly likely that there exist only one candidate object. However, it is separated into several groups in clustering and therefore generates multiple hypotheses with similar poses. The hypotheses is combined into one group and calculate a final pose for this single object.

2) If the overlapping hypotheses belong to different kinds of instances, only one of the hypotheses can be correct. Without any further consideration, The hypothesis with a larger number of correct matches is selected as the recognised object.

3.3 Experiments and Discussions

3.3.1 Consistent Correspondences Search using Graph based Approach

In order to verify the robustness and efficiency of the proposed approach, the following experiment is designed to compared the proposed approach against the most widely used method, RANSAC. Given transformation $[\mathbf{R}, \mathbf{t}]$ and a set of 3D points, transformed points with additional Gaussian noises on each (x, y, z) axis is generated with a standard deviation 0.1 cm. These two sets of 3D points were regarded as correct correspondences. In addition, using the translation vector $\mathbf{t}$, uniformly distributed point sets centred at the origin coordinates $[0., 0., 0.]$ and $[x_t, y_t, z_t]$

are generated. Incorrect matches were generated randomly by associating point pairs between these two uniform distributed points. We were also able to generate multiple sets of consistent correspondences and this set-up can be considered as a single cluster with multiple objects during the object detection step. Two types of experiments were designed to fully verify the effectiveness of the proposed method.

1. **Single consistent set experiment**: The correct set of correspondences had 200 matches and the number of incorrect matches is set as 50, 100, 200 and 400.

2. **Multiple consistent sets experiment**: We have multiple (1, 2 and 3) sets of consistent correspondences in the input raw data. This implies the algorithm is tested under different number of correct matches (i.e. 200, 400 or 600) . The number of outliers is set to be constant as 200.

There are several parameters which are needed to be carefully tuned in our experiments. In both RANSAC and our method, the minimum number of consistent matches in one cluster $\tau_{\mathbf{n}_v}$ was set to be 30 and the threshold of distance error, ω, was set to be 0.15cm which approximately equals to $\sqrt{0.1^2 + 0.1^2 + 0.1^2}$. Given a pair of 3D coordinates and hypothesis $\left[\hat{\mathbf{R}}, \hat{\mathbf{t}}\right]$, if the difference between transformed coordinate and the given coordinate is less than ω, this match is regarded as agreed with $\left[\hat{\mathbf{R}}, \hat{\mathbf{t}}\right]$. In RANSAC, the iteration number was set to 500 irrespective of the number of matches. In each iteration, only 4 pairs of matches were selected to calculate the initial transformation $\left[\hat{\mathbf{R}}, \hat{\mathbf{t}}\right]$[9].

This experiments mainly focuses on estimation accuracy and time consumption. The pose estimation error is measured using the Euler angle instead of the rotation matrix, and the results are shown in table. 3.2. The time consumption of the proposed method is greatly reduced compared to RANSAC. When the number of inliers is 200 and the number of outliers is 400, our method shows larger error as the least square solver is not robust to erroneous data association. The bottom table in Table. 3.2 demonstrates the results where there are multiple object hypotheses in the inlier correspondences. The number of hypotheses increased from 1 to 3 with the number of inliers from 200 to 600. Under this set-up, our approach out-performances RANSAC in both accuracy and timing.

One key parameter in our method is ω. If ω is not tuned to a larger value correctly, the proposed method can easily include incorrect matches which leads to extremely large estimation errors. In order to mitigate this, an additional RANSAC step can be appended with much fewer

TABLE 3.2: Outlier rejection comparative experimental results with RANSAC

(a) Single optimal set searching

Inlier	Outlier	Criterion	Proposed Method	RANSAC
200	50	Rotation Error	$\pm[\mathbf{0.743}, \mathbf{0.059}, \mathbf{0.249}]$	$\pm[2.292, 1.271, 3.759]$
		Translation Error	$\pm[\mathbf{0.014}, \mathbf{0.006}, \mathbf{0.016}]$	$\pm[0.161, 0.071, 0.032]$
		Timing (ms)	$\mathbf{36.54}$	397.43
200	100	Rotation Error	$\pm[\mathbf{0.416}, \mathbf{0.499}, \mathbf{0.716}]$	$\pm[1.303, 2.839, 1.250]$
		Translation Error	$\pm[\mathbf{0.041}, \mathbf{0.011}, \mathbf{0.015}]$	$\pm[0.037, 0.038, 0.159]$
		Timing (ms)	$\mathbf{54.71}$	451.96
200	200	Rotation Error	$\pm[\mathbf{0.236}, \mathbf{0.014}, \mathbf{0.175}]$	$\pm[1.318, 0.583, 1.143]$
		Translation Error	$\pm[\mathbf{0.008}, \mathbf{0.007}, \mathbf{0.003}]$	$\pm[0.041, 0.004, 0.037]$
		Timing (ms)	$\mathbf{91.36}$	632.47
200	400	Rotation Error	$\pm[9.968, 4.431, 6.554]$	$\pm[\mathbf{3.502}, \mathbf{0.091}, \mathbf{1.154}]$
		Translation Error	$\pm[\mathbf{0.022}, \mathbf{0.010}, \mathbf{0.002}]$	$\pm[0.031, 0.050, 0.073]$
		Timing (ms)	$\mathbf{195.29}$	971.60

(b) Multiple optimal sets searching

Inlier	Outlier	Criterion	Proposed Method	RANSAC
200	200	Rotation Error	$\pm[\mathbf{0.236}, \mathbf{0.014}, \mathbf{0.175}]$	$\pm[1.318, 0.583, 1.143]$
		Translation Error	$\pm[\mathbf{0.008}, \mathbf{0.007}, \mathbf{0.003}]$	$\pm[0.041, 0.004, 0.037]$
		Timing (ms)	$\mathbf{91.36}$	632.47
400	200	Rotation Error	$\pm[\mathbf{1.542}, \mathbf{1.859}, \mathbf{0.834}]$	$\pm[1.920, 2.173, 1.728]$
		Translation Error	$\pm[\mathbf{0.341}, \mathbf{0.539}, \mathbf{0.109}]$	$\pm[1.014, 0.836, 0.572]$
		Timing (ms)	$\mathbf{227.43}$	1396.55
600	200	Rotation Error	$\pm[\mathbf{2.031}, \mathbf{1.172}, \mathbf{1.411}]$	$\pm[2.018, 1.554, 1.414]$
		Translation Error	$\pm[\mathbf{0.572}, \mathbf{0.447}, \mathbf{0.710}]$	$\pm[0.549, 0.602, 1.211]$
		Timing (ms)	$\mathbf{468.09}$	3674.12

iteration after the graph based approach. It is still possible to achieve comparable pose estimation accuracy with less computational time. All of the simulation experiments were conducted on 64 bit Ubuntu 12.04 operating system with 2.50GHz Intel(R) Core(TM) i5-2520M CPU.

3.3.2 Multiple Objects Recognition and Pose Estimation

In order to evaluate the performance of the proposed outlier rejection strategy and the accuracy of pose estimation, a cluttered objects dataset is built as shown in Fig. 3.5. In this dataset there are at least 4 different or identical objects in each frame. There are overall 25 different shaped objects in our dataset including cubic objects, cylindrical objects and irregular shaped objects. This experiment compares the proposed framework against the open source MOPED system. In order to verify the feasibility of the system under a similar scenario as Willow dataset, an environment with similar complexity (shown in Fig. 3.7(b)) is constructed where multiple different shaped objects are placed and occlusions are exist.

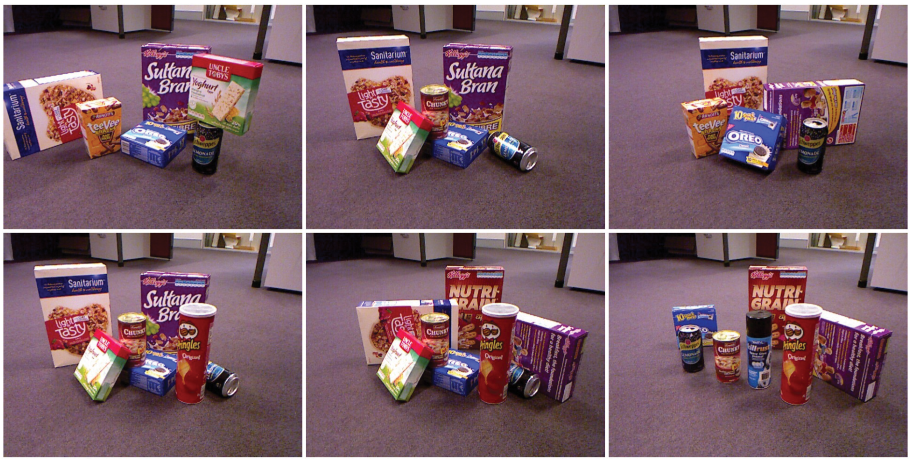


FIGURE 3.5: Sample images from the collected dataset which contains multiple occluded objects.

TABLE 3.3: Object detection results: accuracy, estimate error and time consumption

Method	Precision	Recall	Error(cm)	Time consumption(s)
Proposed system (Graph)	96.54%	85.26%	1.94	0.8405
MOPED	94.20%	83.39%	2.73	1.5473

This subsection presents the results in the cluttered objects dataset which includes 100 frames with a total number of 439 objects that need to be recognised. Our algorithm achieves almost perfect precision in object recognition except in the case where 3 pairs of similar objects are presented. It only fails to recognise 32 objects that are under extreme occlusions among 439 objects. Since it is not able to obtain the accurate relative pose between object and Kinect sensor under this scenario, using the estimated $[\mathbf{R}, \mathbf{t}]$, the re-projection error δ is calculated and used as the measurement of pose accuracy for both translation and rotation. This system is able to achieve comparable precision-recall recognition performance, slightly better pose estimation accuracy. This is expected as 3D information is exploited while MOPED only relies on 2D images only. More importantly, the computational time required for our algorithm is approximately only 50% of the time by MOPED. Following are the key factors for this inferior timing of MOPED.

- RANSAC based plane subtraction generates a mask image which approximately separates the support plane and the target objects. This further decreases the number of extracted features and also speeds up the subsequent steps in Fig. 3.1.

- Compared with 2D clustering in image space, 3D geometrical Mean Shift clustering is much more accurate and robust. Note that these numbers vary depending on the sensor ranges and viewpoints. This also led to the successful implementation of a simpler outlier rejection algorithm.

- The proposed graph based consistent correspondences searching methods is faster than the RANSAC method and achieves robust performances when noisy data are presented.
- Given additional 3D point cloud sensor data, SVD solver is used to obtain the relative pose between two sets of 3D points. Compared with LM optimiser in MOPED, our method is much faster given reliable matches.

Since SIFT extraction is processed using only the CPU in our system which takes almost 80% of the time, much less time consumption (within $400ms$) is anticipated when using GPU computation.

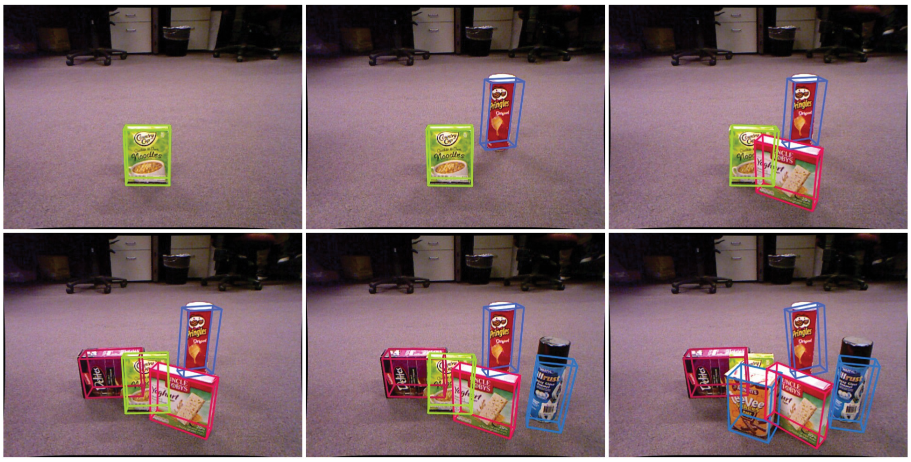
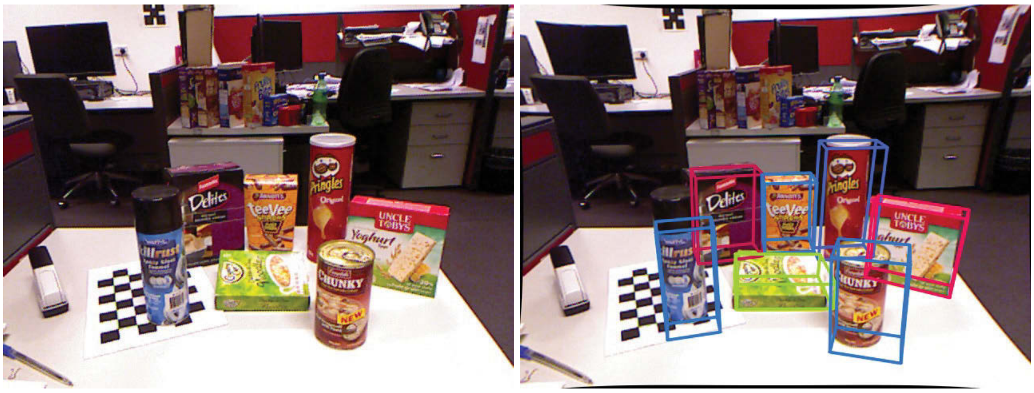


FIGURE 3.6: Objects detection and pose estimation results in cluttered environment.

3.3.3 Discussion

In order to demonstrate the capability of our system on Willow dataset, based on one of the images from Willow dataset (shown in Fig. 3.2), a scenario with similar complexity is built. In Fig. 3.7(b), there are 7 different objects with box shape and cylindrical shape, and the objects are placed in a similar manner as in Fig. 3.2 (7 objects with slightly occluded and one object is placed flat on the plane). Given a closer range and enough number of observed features, shown in Fig. 3.7(b), the method can successfully recognise all the objects with accurate pose. The most notable feature of our proposed system is the time efficiency as discussed in section. 3.3.

The current limitations of our system are listed below:



(a) Constructed scenario similar as Willow dataset

(b) Object detection results from Fig. 3.7(b)

FIGURE 3.7: Object detection results in a similar scenario as Willow dataset.

- depth information is not fully used in the recognition step. Therefore the recognition performance is limited by the keypoint features such as SIFT;
- the current system is not able to perform well under a highly cluttered environment especially when an inadequate number of features are observed;

The first problem will be solved by using RGB-D features which will be discussed in the next chapter and the second issue motivates the main focus of this PhD work, active object detection, which will be presented in Chapter. 6.

3.4 Summary

This chapter presents a textured object recognition and pose estimation pipeline using an RGB-D sensor in a highly cluttered environment. A graph-based consistent correspondences search algorithm is proposed by exploiting the relative geometric constraints between each pair of matches. Our system benefits by completely using the depth information in pose estimation by directly using 3D-3D correspondences. This approach is faster to execute and also leads to higher accuracy. Combined with other improvements on each step and comprehensive experiments, it is shown that the proposed system performs well under a number of different scenarios in an indoor environment and is capable of reducing 50% of the computational time compared to MOPED. However, this chapter fail to exploit the depth information in the recognition phase thus cannot detect less-textured objects or objects under severe illumination conditions.

Chapter 4

A Novel RGB-D Feature for Challenging Object Detection

Chapter. 3 presents a framework for detection and pose estimation of textured object. However, in daily life, there also exist various kinds of objects that do not possess rich texture information. Besides, one of the limitations of the framework for detection and pose estimation of textured object is that its object detection results are subjected to the illumination conditions. In order to guarantee a reliable feature extraction and matching performance and to further ensure robust object detection results, the illumination conditions have to be good enough. To address these two issues, this chapter presents a novel appearance and shape feature, RISAS, which is robust to viewpoint, illumination, scale and rotation variations.

RISAS consists of a keypoint detector and a feature descriptor both of which utilise texture and geometric information present in the appearance and shape channels. It is robust to illumination variations and also capable of extracting salient keypoints from geometric information. A novel response function based on the surface normals is used in combination with the Harris corner detector for selecting keypoints in the scene. A strategy that uses the depth information for scale estimation and background elimination is proposed to select the neighbourhood around the keypoints in order to build precise invariant descriptors. Proposed descriptor relies on the ordering of both grayscale intensity and shape information in the neighbourhood. Comprehensive experiments which confirm the effectiveness of the proposed RGB-D feature when compared with CSHOT [142] and LOIND[51] are presented. Furthermore, the utility of incorporating texture and shape information in the design of *both* the detector and the descriptor is highlighted by demonstrating the enhanced performance of CSHOT and LOIND when combined with the RISAS detector. RISAS has been adopted for the framework presented in Chapter. 3 to evaluate the object detection performance under extreme illumination conditions.

This chapter is structured as follows: The methodology of the proposed RGB-D feature is illustrated in Section. 4.1 including the keypoint detector(4.1.1) and feature descriptor(4.1.2); the proposed feature, RISAS, is validated thoroughly using a public dataset and a specifically designed RGB-D feature evaluation dataset in Section. 4.2. The object detection results are demonstrated in Section. 4.3 and Section. 4.4 concludes this chapter.

4.1 A Novel Rotation, Illumination and Scale invariant RGB-D Feature

This section describes the proposed Rotation, Illumination and Scale invariant Appearance and Shape feature, RISAS, in detail. RISAS is built on the previous work about an illumination and scale invariant RGB-D descriptor called Local Ordinal Intensity and Normal Descriptor (LOIND) [51]. The detector and descriptor are explained in detail in Section.4.1.1 and Section.4.1.2.

4.1.1 Keypoint Detector

The main advantage of using depth information in keypoint detection is the fact that information rich regions in the depth channel are also given due consideration without being ignored when these regions lack texture information. Both the proposed detector and the descriptor use similar information and thus are tightly coupled giving rise to superior matching performance.

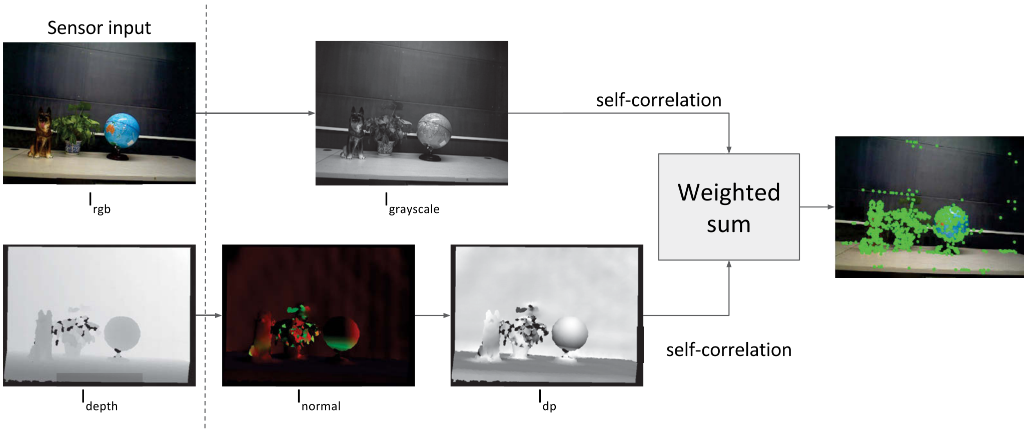


FIGURE 4.1: Algorithm flowchart of the keypoint detector in the proposed RISAS RGB-D feature.

The flowchart of the keypoint detection method is shown in Fig. 4.1 where I_{rgb} is the original RGB image and $I_{\text{grayscale}}$ is the converted grayscale image. I_{normal} is the 3 channel normal vector image and I_{dp} is the dot product image. The key steps are listed below

1. For each point in the depth image I_{depth}, the surface normal vector is calculated. From the three components of the normal vector, the corresponding normal image I_{normal} with three channels is created.

2. Using I_{normal}, the three angles $[\alpha, \beta, \gamma]$ between each normal vector and the $[x, y, z]$ axis of the camera coordinate system are computed. The angle range $[0, \pi]$ is segmented into n_s sectors labelled with $[1, ..., n_s]$ and each computed angle is mapped into one of these sectors. In this work, n_s is set to be 4 as shown in Fig. 4.2. For example, normal vector $\mathbf{n} = \left[\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}\right]$ has the $[\alpha, \beta, \gamma] = [54.7°, 54.7°, 54.7°]$ will be labelled as $[2, 2, 2]$;

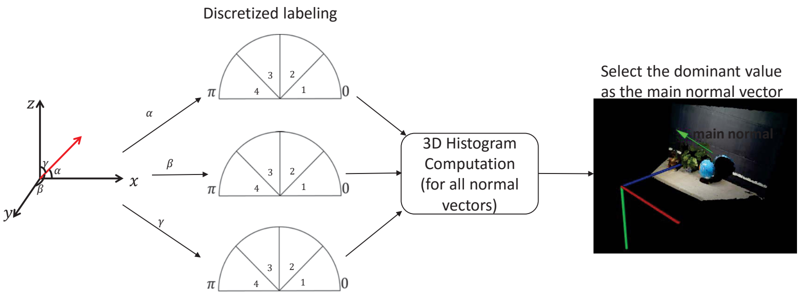

Discretized labeling
Select the dominant value as the main normal vector
z
x
y
α
β
γ
π
0
1
2
3
4
3D Histogram Computation (for all normal vectors)
main normal

FIGURE 4.2: Calculating the main normal vector of the depth image to extract RGB-D keypoints.

3. Using this labelled image, a statistical histogram is built to capture the distribution of labels along each channel. From this histogram, the highest entry for each channel is chosen and the corresponding label $[n_X, n_Y, n_Z]$ is used to represent the most frequent label where $n_X, n_Y, n_Z \in \{1, ..., n_s\}$. Using these three values, the "main" normal vector $\mathbf{n}_{main}$ of the depth image I_{depth} is defined.

4. Calculate the dot-product between $\mathbf{n}_{main}$ and each normal vector in I_{normal}. This describes the variation of information in the depth channel. The dot product value is then normalised into range $[0, 255]$. Using this value, a novel dot-product image I_{dp} is created which is approximately invariant to the viewpoint of the sensor.

5. The similar principle as in the Harris detector is adopted to compute the response value $E(u, v)$ using the grayscale image $I_{\text{grayscale}}$ and the dot product image I_{dp}. The response

value is thresholded to select points that show an extreme value in the weighted sum of two response values from $I_{\text{grayscale}}$ and I_{dp}, as shown in (4.1):

$$\begin{aligned} E\left(u,v\right) = \sum_{x,y} \omega(x,y)[\tau\left(I(x+u,y+v) - I(x,y)\right)^2 \\ + \left(1-\tau\right)\left(P(x+u,y+v) - P(x,y)\right)^2] \end{aligned} \tag{4.1}$$

where (u,v) is the keypoint coordinate in image space and $\omega(x,y)$ is the window function centred at (u,v) which is a Gaussian function in the work presented in this thesis. $I(u,v)$ is the intensity value at (u,v) and $P(u,v)$ is the normalized dot product value at (u,v). Empirical study shows that τ plays a critical role in balancing appearance information and geometric information in keypoint detection. Because of the fact that an rgb/grayscale image is more information rich compared with a depth image and provides more variations, τ should assign a larger value to the rgb image. Fig. 4.3 provides precision-recall curves for different τ value for the same scenario. τ is set empirically as 0.8.

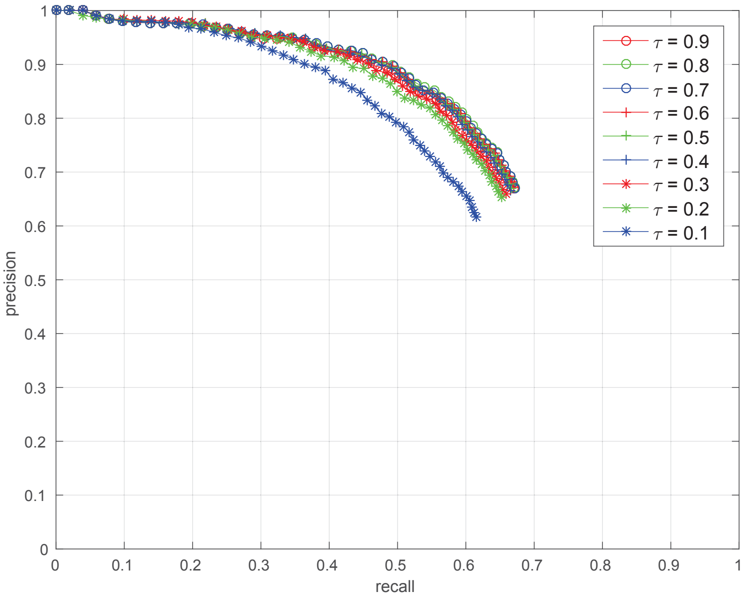


FIGURE 4.3: Precision-Recall curves for difference τ value in feature point extraction.

This strategy clearly identifies keypoints from regions that are information rich in both appearance and geometry.

4.1.2 Feature Descriptor

Scale Estimation and Neighbourhood Region Selection

For grayscale images, the scale of the keypoint is estimated by finding the extreme value in scale space using image pyramid. Typical examples are as SIFT[95] and SURF[17]. With the development of modern RGB-D sensors such as Kinect and Xtion, the scale can be easily measured using the depth information captured from the sensor. In both LOIND[51] and BRAND[107], the following empirical equation scales the distance range between $[2, 8]$m into scale range $[1, 0.2]$ in a linear relationship. Scale value for distance less than 2m is truncated to 1.

$$s = \max\left(0.2, \frac{3.8 - 0.4\max(2, d)}{3}\right) \tag{4.2}$$

After s is estimated, the neighbourhood region that is used to build the descriptor is selected with radius R in a linear relationship with scale value s, as shown in [51, 107]. A critical deficiency in their approach is that the neighbourhood region is selected without considering the geometric continuity. In the following, a more accurate method is presented for selecting the neighbourhood region from which the descriptor is built.

1. Based on (4.2), initial value of the scale s is estimated. The radius R of the patch is computed using (4.3) which was derived using extensive experimentation.

   $$R = \left(-5 + 25 * \min\left(3, \frac{\max(0.2.s_{\max})}{\max(0.2.s_{\min})}\right)\right) \cdot s \tag{4.3}$$

   where $s_{\max}$ and $s_{\min}$ are the maximum and minimum scale values in the image. If scale varies gently in the neighbourhood region, a smaller R is a better option and vice versa. The patch centred at keypoint k_i in 2D image space is denoted as $\mathbf{P}^{uv}(k_i)$ and the corresponding patch in 3D point cloud space is represented as $\mathbf{P}^{xyz}(k_i)$;

2. For each point $p \in \mathbf{P}^{xyz}(k_i)$, the outlier neighbouring points from the keypoint k_i are removed according to (4.4). The main objective of this step is to remove the points pertaining

to the background . This step of eliminating the background was found to produce significant improvements in the matching performance.

$$f(p) = \begin{cases} 1 & \text{if } \|p - k_i\| < t \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

where t is the threshold and set to be 0.1 meter in this work. Only the neighbouring points with $f(p) = 1$ are kept;

3. Ellipsoid fitting is conducted for the processed 3D neighbouring points $\bar{\mathbf{P}}^{xyz}(k_i)$ based on the following equation.

$$\frac{(x - x_{k_i})^2}{a^2} + \frac{(y - y_{k_i})^2}{b^2} + \frac{(x - z_{k_i})^2}{c^2} = 1 \tag{4.5}$$

where a, b and c are the length of the axes. The 3D ellipsoid is projected into the image space for the new accurate patch $\bar{\mathbf{P}}^{uv}$ with radius $\bar{R}$ for further descriptor construction.

Fig. 4.4 demonstrates the results of the proposed neighbourhood selection method. The default strategy (left) selects the whole region (shown in red) which covers both foreground and background area. However, the background points have an adverse effect on the local descriptor. Our approach (right) eliminates the background points (shown in blue) and constructs the descriptor using the foreground (shown in red) only, leading to more robust descriptor matching performances.

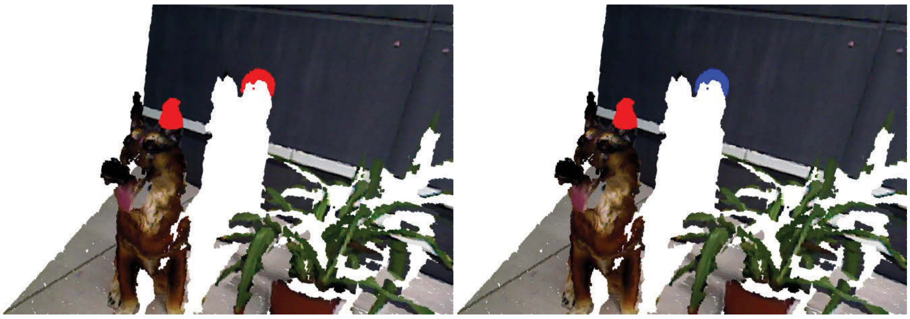


FIGURE 4.4: Comparison results for different neighbourhood selection methods for descriptor construction.

Orientation Estimation

In LOIND[51], the dominant orientation θ of the selected patch is computed from the depth information only. Although it works reasonably well under different scenarios it is sensitive to the noise in neighbourhoods where the normal vectors are similar to each other. An alternative novel dominant orientation estimation algorithm is proposed which is more robust and efficient compared with LOIND[51]:

1. Given the processed 2D patch $\bar{\mathbf{P}}^{uv}$ and 3D patch $\bar{\mathbf{P}}^{xyz}$, PCA is adopted to compute the eigenvalues $[e_1, e_2, e_3]$ (in *descending* order) and corresponding eigenvectors $[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$.

2. Given the eigenvectors $[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$, the 3D dominant orientation d_{3D} of the patch is computed as follows:

$$d_{3D} = \begin{cases} \frac{\mathbf{v}_1 \times \mathbf{v}_2}{|\mathbf{v}_1 \times \mathbf{v}_2|} & \text{if } (e_2 > \gamma e_1) \wedge (e_3 \leq \gamma e_1) \\ \text{rejected} & \text{if } (e_2 > \gamma e_1) \wedge (e_3 > \gamma e_1) \\ \mathbf{v}_1 & \text{others } (e_1 \text{ is signficantly larger}) \end{cases} \tag{4.6}$$

   where γ is set within $[0, 1]$. If the e_1 is significantly larger than the other two, the 3D dominant orientation is set to be the corresponding eigenvector $\mathbf{v}_1$. If e_2 is close to e_1, both eigenvector $\mathbf{v}_1$ and $\mathbf{v}_2$ are considered in computing the dominant orientation by taking the cross-product of these two vectors. Further if both e_3 and e_2 are closer to e_1 which means no clear differences between the 3 eigenvalues, this keypoint is rejected because the depth channel will not be able to provide distinctive information. Threshold γ determines when the second eigenvalue e_2 can be regarded as "close" enough to the largest eigenvalue e_1 which is set to be 0.8 through experiments.

3. Project the 3D dominant direction d_{3D} into the image plane and get the 2D dominant direction d_{2D}. We use θ to denote the angle between d_{2D} and u axis in image space.

Descriptor Construction

Based on the results from the above steps, the descriptor of keypoint $k_i = [u, v]$ can be constructed using the neighbourhood region with radius R and the angle θ following the main ideas used in LOIND[51]. The descriptor is based on the relative order information in both grayscale and depth channels. The descriptor is constructed in a three-dimensional space, as shown in Fig. 4.5 below where $[x, y, z]$ axes denote the spatial labelling, the intensity labelling and the angles labelling respectively.

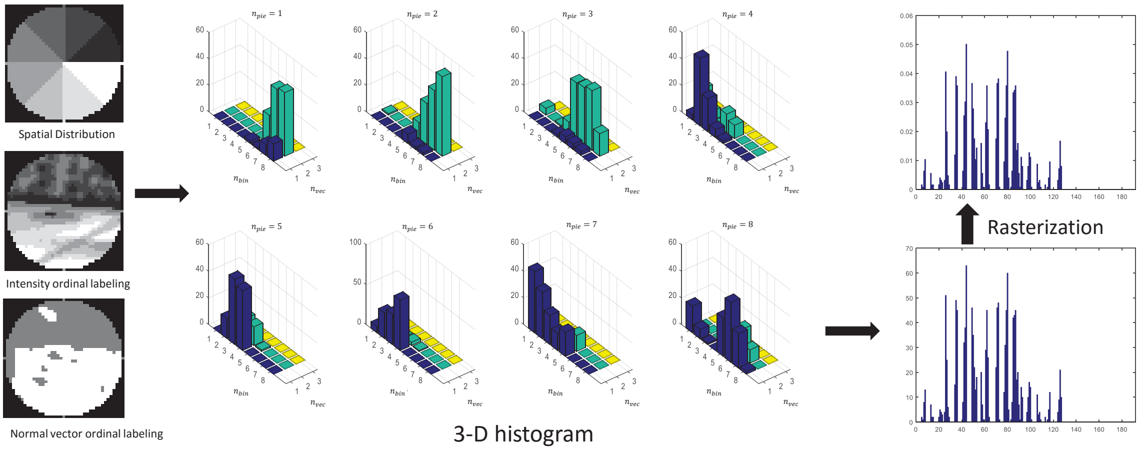


FIGURE 4.5: Algorithm flowchart of constructing descriptor of the RGB-D feature.

- **Encoding Spatial Distribution**

  For spatial distribution, the pixels in the region (u, v, R, θ) are labeled based on n_{pie} equal-size spatial sectors. Larger the number of sectors, the more discriminative the descriptor, but this clearly effects timing for both construction and matching.

- **Encoding Grayscale Information**

  In order to enhance illumination invariance, instead of constructing the descriptor in the absolute intensity space, the statistical histogram is built using the relative intensity with respect to the intensity value of the keypoint. According to the rank of all the pixels in the patch, the intensity values are grouped into n_{bin} equally sized bins. For example, given 100 intensity levels and 10 bins, each bin has 10 intensity levels (i.e., orderings of $[1, 10], [11, 20], \ldots, [91, 100]$).

- **Encoding Geometrical Information**

  Given the normal vector of each point, the dot product between the normal vector of the selected keypoint $\mathbf{n}_{p_k}$ is computed firstly and the normal vector of each point in the neighbourhood patch $\mathbf{n}_{p_i}$.

$$\rho_i = |\langle \mathbf{n}_{p_k}, \mathbf{n}_{p_i} \rangle| \tag{4.7}$$

> Due to the fact that normal vectors from small patches are similar to each other, the distribution of ρ_i is highly unbalanced where the majority of ρ_i fall into the range close to 1. $\bar{\rho}$ is set as 0.9 and any $\rho_i \geq \bar{\rho}$ are grouped into one category. The remaining dot products are ranked and grouped into n_{vec} bins. Points are then labelled based on the group they belong to. Therefore, in normal vector space encoding, there are overall $n_{vec} + 1$ labels.

During the empirical study, 12 different combination of parameters $n_{pie} = \{4, 8, 12\}, n_{bin} = \{8, 16\}$ and $n_{vec} = \{1, 2\}$ are tested. The precision-recall curves are presented in Fig. 4.6. Considering both performance and efficiency, in the experiments section, parameters are set as $n_{pie} = 8, n_{bin} = 8, n_{vec} = 2$ thus constructing a 192-dimensional ($\dim = n_{pie} \cdot n_{bin} \cdot (n_{vec} + 1)$) descriptor.

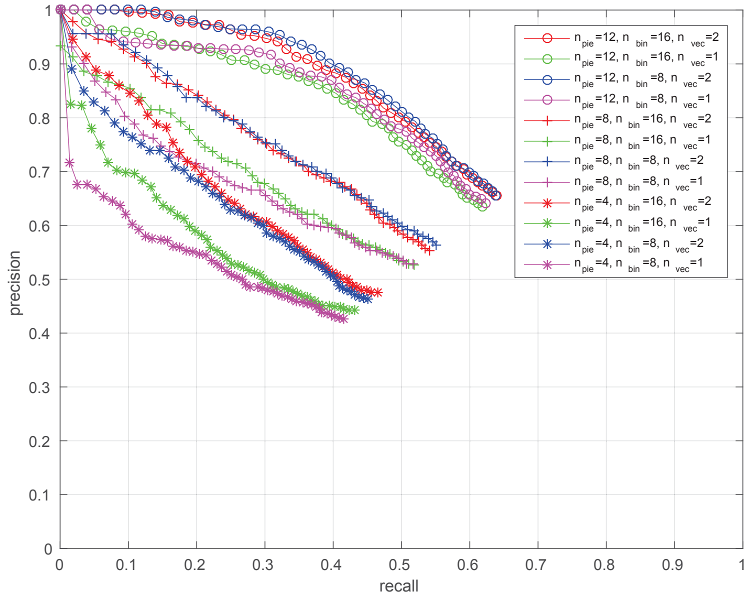


FIGURE 4.6: Precision recall curves under different parameterisation of descriptor construction.

4.2 Experimental Results

This section presents the performance comparison of RISAS against CSHOT, LOIND and other methods. In order to highlight the merits of using both appearance and depth channels, the results of comparative experiments using SIFT features have been reported. In these experiments, a public RGB-D dataset was used which was originally designed for object detection[1]. This dataset does not include examples of rotation, scale or illumination changes independently and therefore

[1] `http://rgbd-dataset.cs.washington.edu/`

it is unable to fully illustrate the effectiveness of the RISAS in such situations. Therefore a new dataset has been developed for further detailed evaluations[2].

4.2.1 Evaluation Method

Firstly, keypoints from two frames are extracted and the descriptors are constructed for all these keypoints. Nearest Neighbour Distance Ratio (NNDR) is used to establish the correspondences of keypoints between a pair of images. The reprojection error is used to determine whether a correspondence is correct using the equation below:

$$||p_i - (\mathbf{R}p_j + \mathbf{t})\,|| \leq d_{\min} \tag{4.8}$$

where p_i and p_j are 3D points from frames i and j. $\mathbf{R}$ and $\mathbf{t}$ denote the true rotation and translation and are given during the evaluation. If the re-projection error is less than $d_{\min}$(set to be 0.05 m), the match is regarded as a correct one. In the next subsection, the *percentage of inliers* is used to describe the invariance of the features w.r.t scale variations. *Precision-Recall* curves are used to evaluate the performance of the RGB-D features under other types of variations similar to [100] .

4.2.2 Experimental Results and Analysis

In order to evaluate the performance of the proposed RISAS feature, the following comparative experiments have been conducted :

1. 3D ISS keypoint detector and RGB-D CSHOT descriptor: ISS has been combined with different 3D descriptors for evaluation in Guo et al.'s survey[60]. Implementations of these in PCL[128] were used in the experiments presented in this section.

2. Uniform sampled keypoints and RGB-D CSHOT descriptor: Uniform sampling method for keypoint detection was used in Aldoma et al.'s work [6] for 3D object recognition[3] In these experiments, the uniform sampling method was adopted and the methods provided in PCL were used.

3. 2D SIFT keypoint detector and RGB-D CSHOT descriptor: Publically available implementation of SIFT detector from VLFeat[152] and CSHOT descriptor from PCL[128] were used.

[2] This dataset can be downloaded from `http://kanzhi.me/rgbd-descriptor-dataset/` to make it possible for the community to use this in future research

[3] Random sampling is used in the SHOT[144] paper and CSHOT paper[142].

This was used as an example of a combination of a 2D keypoint detector and a RGB-D descriptor.

4. Proposed RISAS keypoint detector and RGB-D CSHOT descriptor: Matlab implementation of the RISAS detector together with the PCL implementation of CSHOT was used.

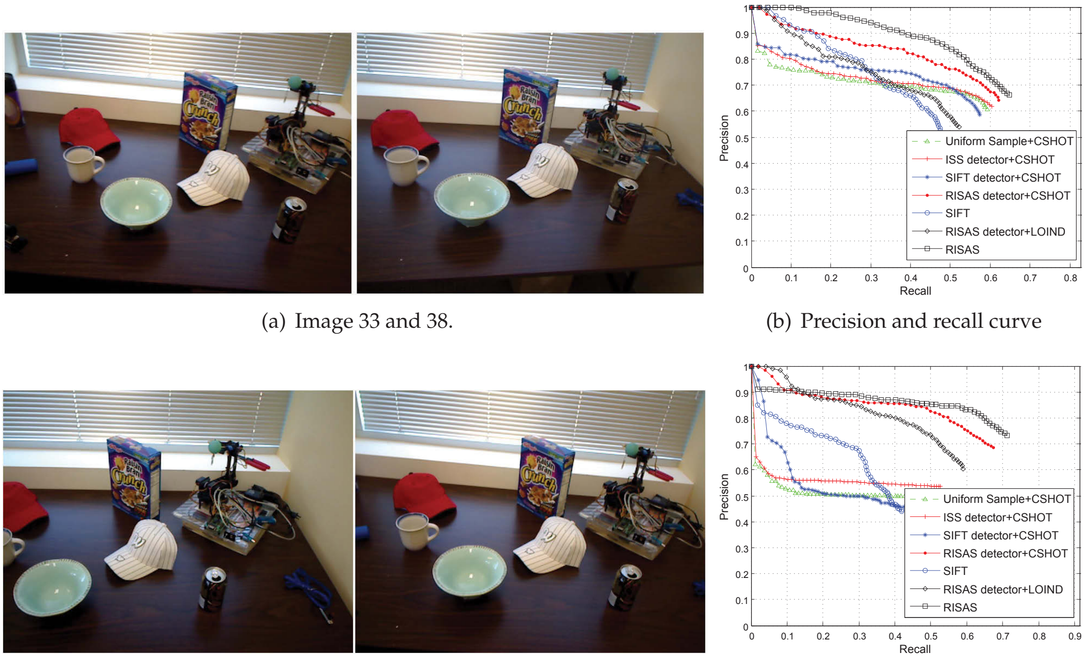
5. 2D SIFT feature (detector and descriptor) as implemented in VLFeat.

6. Proposed RISAS keypoint detector and LOIND descriptor that were implemented in Matlab.

All of the experiments were performed on a standard desktop PC equipped with an Intel i5-2400 CPU.

4.2.3 Object Recognition Dataset

The information-rich sequence *table_1* from the RGB-D scene dataset [85] was chosen and some of the results are presented in Fig. 4.7. As the figure indicates, RISAS and the combination of RISAS detector and CSHOT descriptor show a significantly larger area under the curve and thus demonstrate the best performance.



(a) Image 33 and 38.

(b) Precision and recall curve

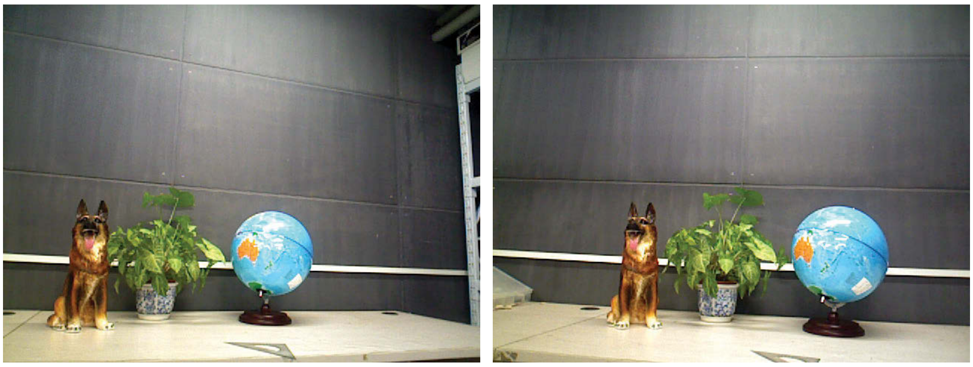
(c) Image 25 and 32.

(d) Precision and recall curve

FIGURE 4.7: Evaluation results on RGB-D scene dataset.

4.2.4 RGB-D Feature Evaluation Dataset

In the constructed dataset, the following four common variations were considered: 1) viewpoint, 2) illumination, 3) scale and 4) rotation.



(a) Image 12

(b) Image 24

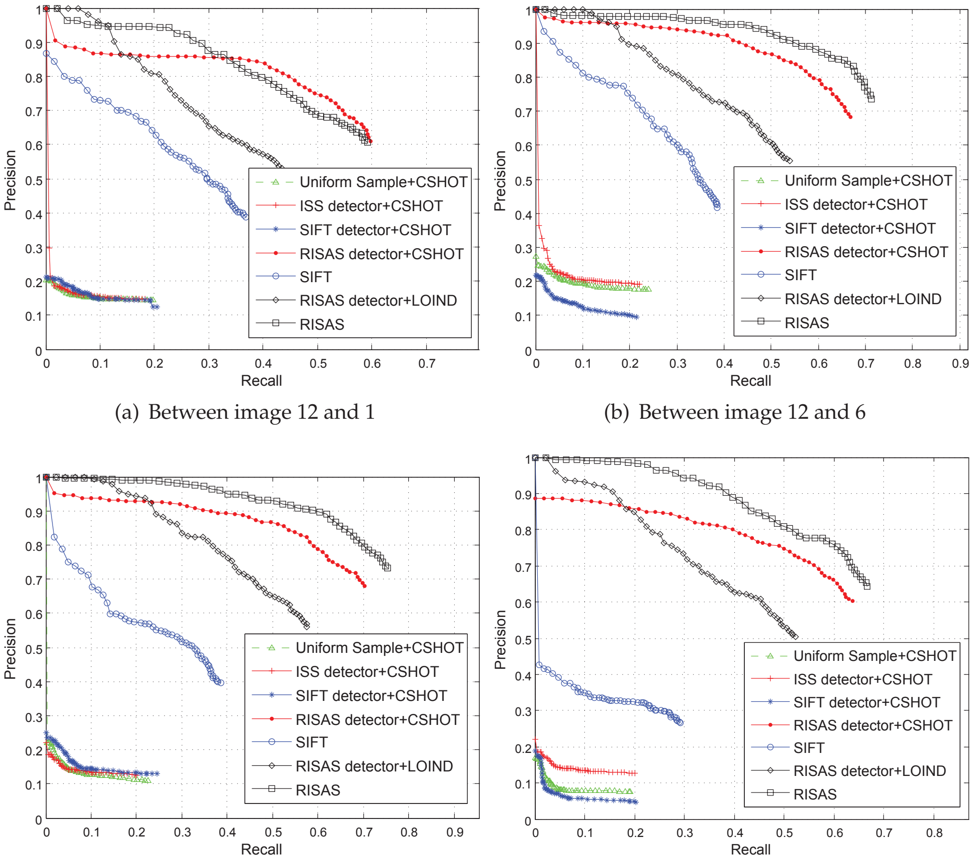
FIGURE 4.8: Example images under different viewpoint variations.

Viewpoint Invariance

24 images have been collected by moving the sensor around the objects in approximately 60° at 0.7 meters away from the objects. The angle between each pair of consecutive frames is approximately 3°. In order to estimate the true transformation between each pair of frames and to further evaluate the performance of descriptors, the RGBD-SLAM[46] scheme has been adopted. The RBG-D algorithm computes the optimised poses and these optimised poses have been regarded as the ground-truth. The image which faces straight forward to the object (in the middle with index 12) was selected as the reference image, and it was matched with two images one on the left and the other one on the right side (with indices 1, 6, 18 and 24) to the reference one. Image 12 and 24 are presented in Fig. 4.8. The Precision-Recall curves of these four pairs of images are shown in Fig. 4.9. RISAS is significantly superior compared with all other methods. CSHOT performs well when used with the RISAS detector while performing surprisingly poorly with SIFT and ISS detectors, and also with uniform sampling. It also has been noticed that SIFT doesn't perform as expected under these scenarios with approximate 30° of viewpoint change.

Illumination Invariance

In order to validate the performance of RISAS under illumination variations, a dataset was constructed which consists of five different levels of illumination conditions: 1) square 2) square root



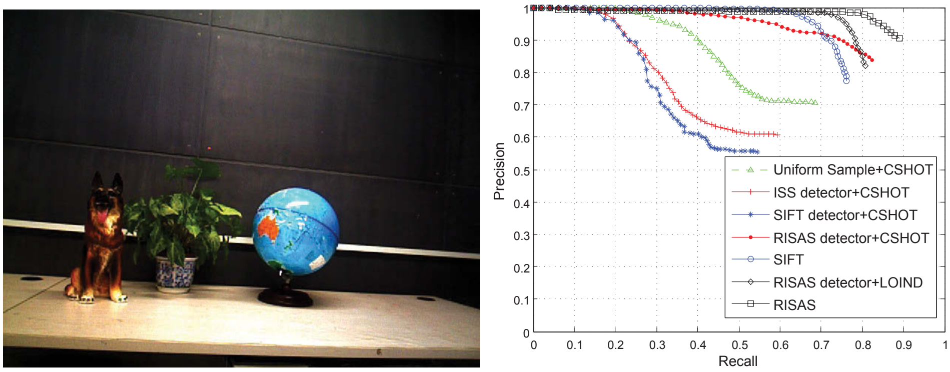
(a) Between image 12 and 1

(b) Between image 12 and 6

(c) Between image 12 and 18

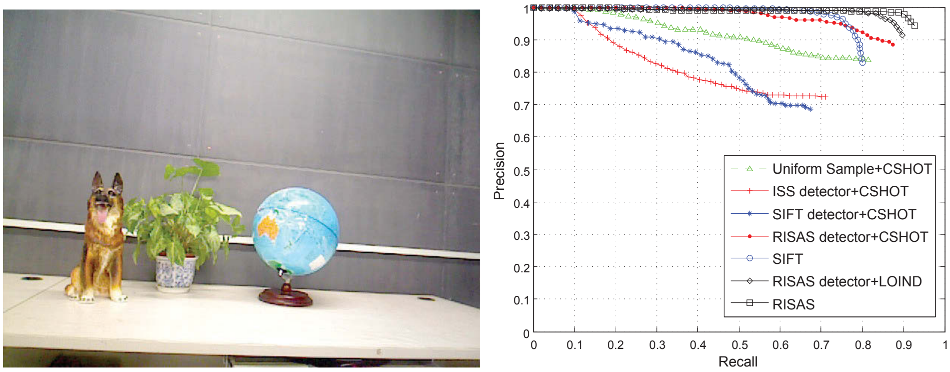
(d) Between image 12 and 24

FIGURE 4.9: Precision-Recall curves under different viewpoint variations.



(a) (b)

FIGURE 4.10: Example images and results under square root illumination variation



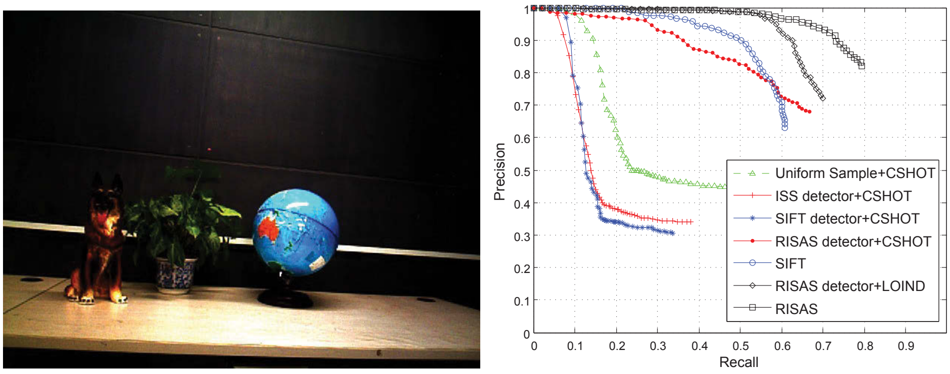
(a) (b)

FIGURE 4.11: Example images and results under square illumination variation

3) cube, 4) cube root and 5) natural illumination variation, as shown in the left column from Fig. 4.10 to Fig. 4.14. The reference image is shown in Fig. 4.15. As the figures demonstrate, the proposed RISAS feature shows the best performance compared with other approaches, i.e. the precision value of RISAS is almost equal to 1.0 when the recall value is 0.7 regardless of the extent of the illumination variation. It is interesting to note that SIFT performs quite well while at the same time performance of CSHOT is significantly enhanced by using it together with the RISAS detector.

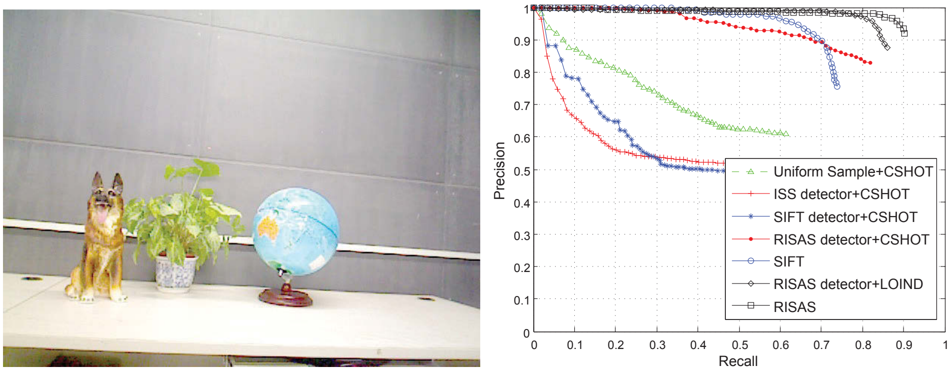
Scale Invariance

In this experiment, 10 images were collected with the variations in z axis of the sensor coordinate system. The first frame that was captured at 1.1 m from the object was selected as the reference image and all other images were captured by moving the camera backwards in steps of 0.1 m. A



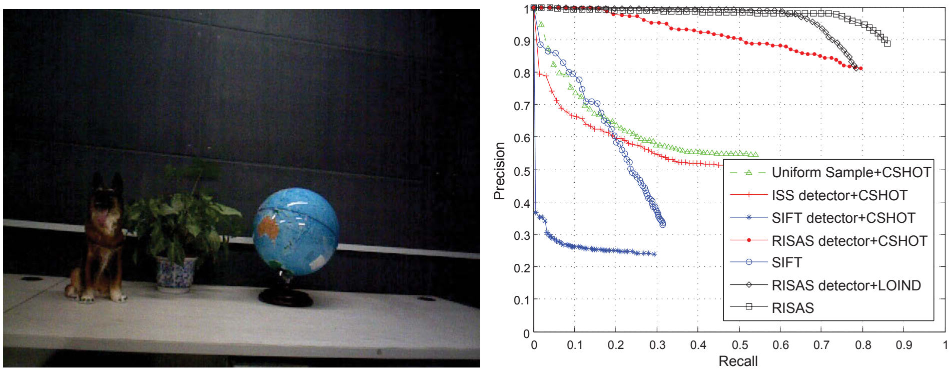
(a) (b)

FIGURE 4.12: Example images and results under cube root illumination variation



(a) (b)

FIGURE 4.13: Example images and results under cube illumination variation



(a) (b)

FIGURE 4.14: Example images and results under illumination change using ND mirror

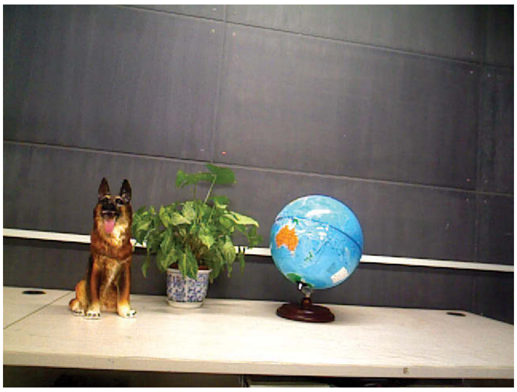
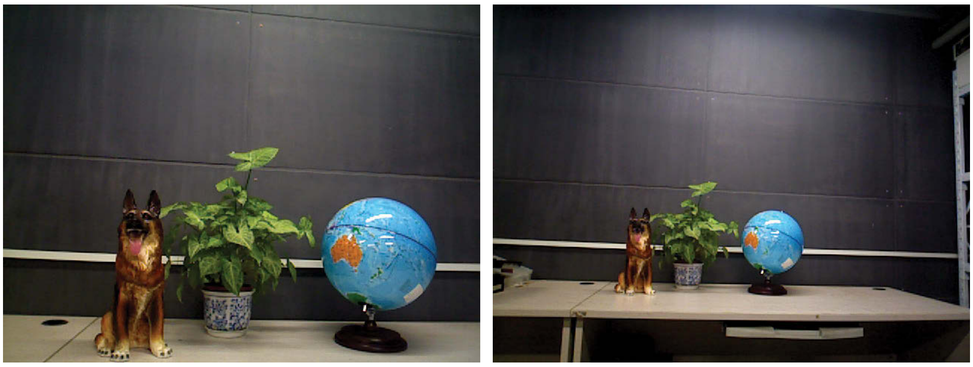


FIGURE 4.15: Reference image for illumination and rotation variation.

pair of images of scale variations is shown in Fig. 4.16 and the matching accuracy w.r.t the scale variation is shown in Fig. 4.17. While RISAS gives the best performance, RISAS detector used with CSHOT also demonstrates good results. All the other methods are significantly inferior.



(a) Original image as reference, captured at distance $\approx 1.1m$

(b) Image captured at the distance $\approx 1.9m$

FIGURE 4.16: Example images under different scale variations.

Rotation Invariance

Fig. 4.18 demonstrates the performance of RISAS under 3D rotation. The reference image is shown in Fig. 4.15 for illumination variations. Precision-recall curves are presented in Fig. 4.19. Under 3D rotations, RISAS and the combination of RISAS detector and CSHOT achieve the best results.

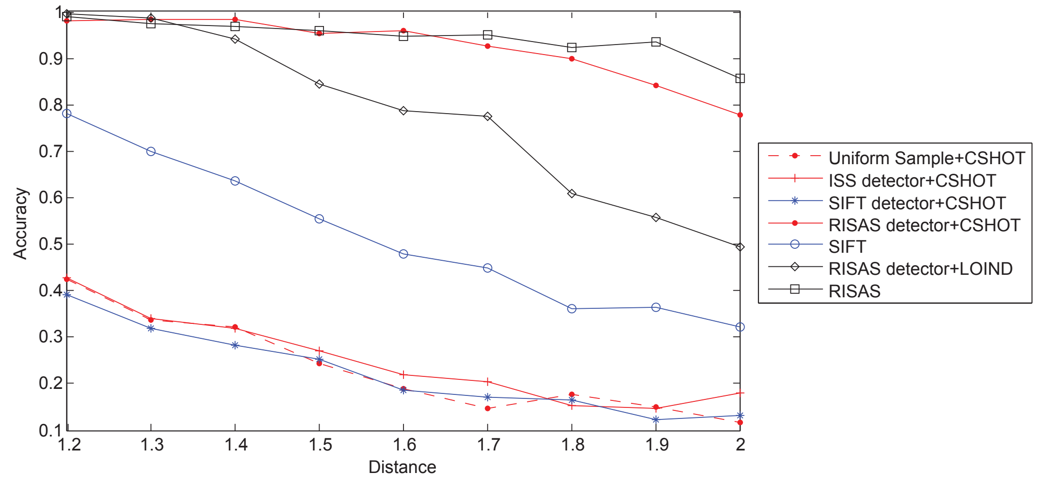


FIGURE 4.17: Comparative matching results under different scale variations.

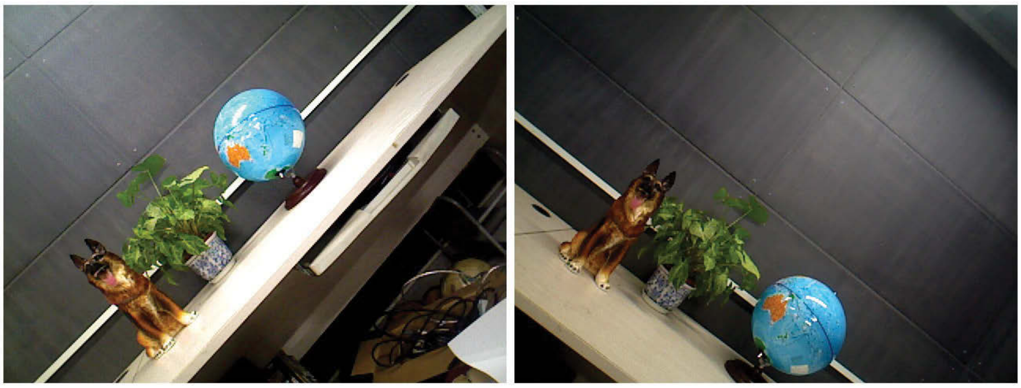
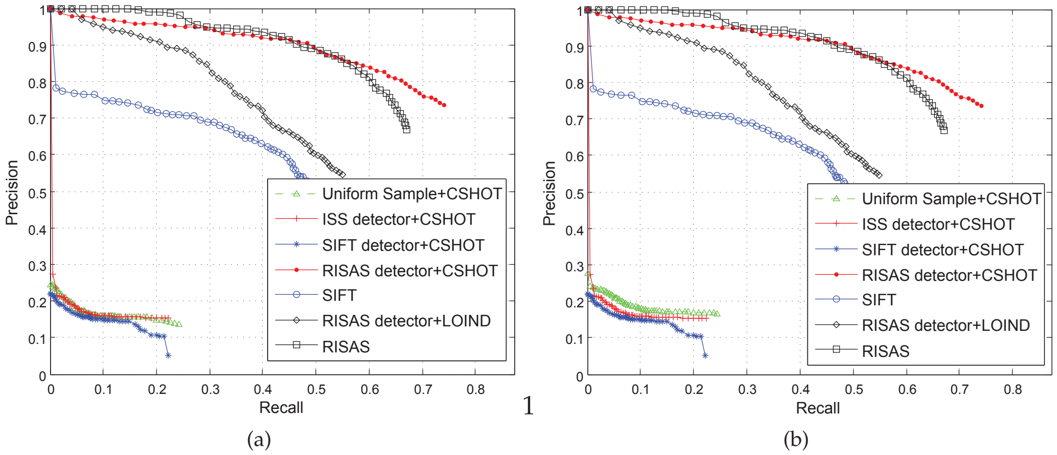


FIGURE 4.18: Example images of different 3D rotations.



(a) (b)

FIGURE 4.19: Precision-Recall curves corresponding to Fig. 4.18.

Discussion

Results from the experiments show that, overall, RISAS provides the best results when compared with other approaches. RISAS shows clear advantages over the other methods under viewpoint variations. Under illumination variations, RISAS outperforms other methods significantly except for LOIND where LOIND achieves comparable results. Under scale and rotation variations, RISAS and the combination of RISAS detector and CSHOT descriptor demonstrate the best performance.

It is clear that using the RISAS detector with CSHOT significantly enhances its matching performance. This confirms that a suitable RGB-D detector is critical for the performance of a RGB-D descriptor. In RISAS, the descriptor performs well if the neighbourhood of the keypoint shows higher normal vector variations. This variation is precisely the main consideration in developing the detector.

In the current unoptimised Matlab based implementation, RISAS takes 20 seconds to complete both keypoint detection and descriptor construction for an image 640×480 captured from Kinect/Xtion. On the same PC with C/C++ implementations in PCL [128], ISS[163] takes nearly 6 seconds and CSHOT takes almost 1 second to process a similar frame. Computational time of RISAS can be significantly improved when implemented in C/C++.

4.3 Object Recognition using RISAS

First, Section. 4.3.1 briefs the pipeline of the recognition framework using the proposed RISAS feature for object recognition tasks. Its superior performance especially under extreme illumination variations and also with less-textured objects are later demonstrated through experiments presented in Section. 4.3.2 and Section. 4.3.3.

4.3.1 Pipeline in Brief

A simple object recognition pipeline shown below is adopted which detects target objects using 2D-2D feature correspondences only.

1. **Feature extraction**: extract RISAS features on cropped object image I_{object} and the whole image which contains the target object, I_{scene};

2. **Feature matching**: the extracted features from I_{object} and I_{scene} are matched using ANN algorithm and ratio test is applied with the ratio equals 0.7;

3. **Matches filtering**: affine transformation is used as a criterion to calculate the transformation related to the matched features and to remove the outliers;

4. **2D Object Localisation**: the cropped image I_{object} is transformed using the estimated affine transformation into the I_{scene} thus indicates the detected boundingbox of the object;

4.3.2 Object Recognition in extreme Illumination Conditions

The object image I_{object} is shown in Fig. 4.20 and the detection results are shown in Fig. 4.21. The upper row shows the detection results where the illumination level of the images is set to be 40% of the original image and the bottom row shows the detection results where the illumination level is set to 160% of the original image. The testing images are generated using the equation below where $\alpha = 0.4$ or 1.6. $I_{\text{normalised}}$ is the normalised image where each value is in range $[0, 1]$.

$$I_{\text{illu}} = \alpha I_{\text{normalised}} + \mathcal{N}(0, 0.1) \tag{4.9}$$

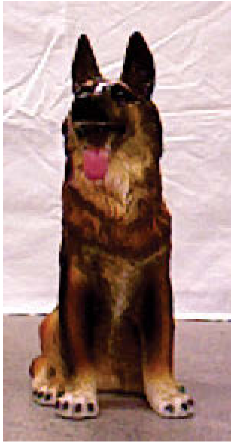


FIGURE 4.20: I_{object} for object recognition under illumination variations.

4.3.3 Less-Textured Object Recognition

In this subsection, the effectiveness of the proposed RGB-D feature, RISAS, is validated for less-textured object detection using a kettle as an example, as shown in Fig. 4.22. The kettle has nearly no texture information on its surface but shows rich geometric information. In this experiments,

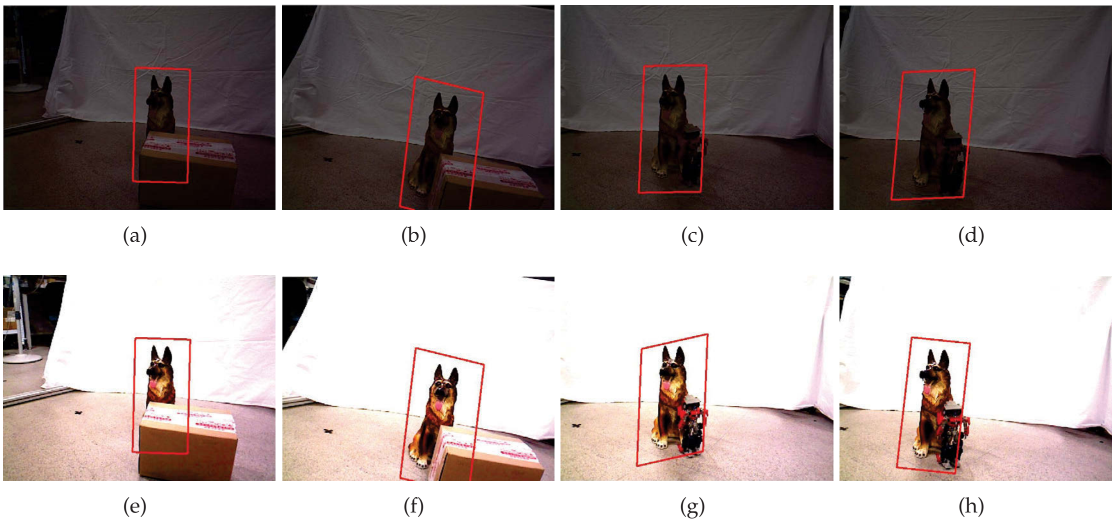


FIGURE 4.21: Object detection results corresponding to the object in Fig. 4.20.

in both detector and descriptor, larger weights are assigned to depth channel in order to achieve better performance.

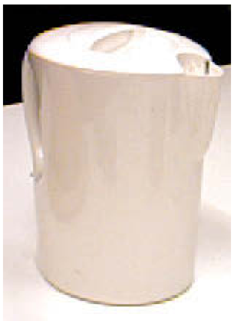


FIGURE 4.22: I_{object} for less-textured object detection.

The detection results are shown in Fig. 4.23. In most of the time, the object can be detected correctly. In Fig. 4.23(e), due to inaccurate feature detection results, there is a significant offset in the detection results. However, this is understandable for object detection under viewpoint variations.

4.4 Conclusion

This chapter presents a novel RGB-D feature (RISAS) which consists of a tightly coupled RGB-D keypoint detector and a descriptor. A novel 3D representation, dot-product image is combined

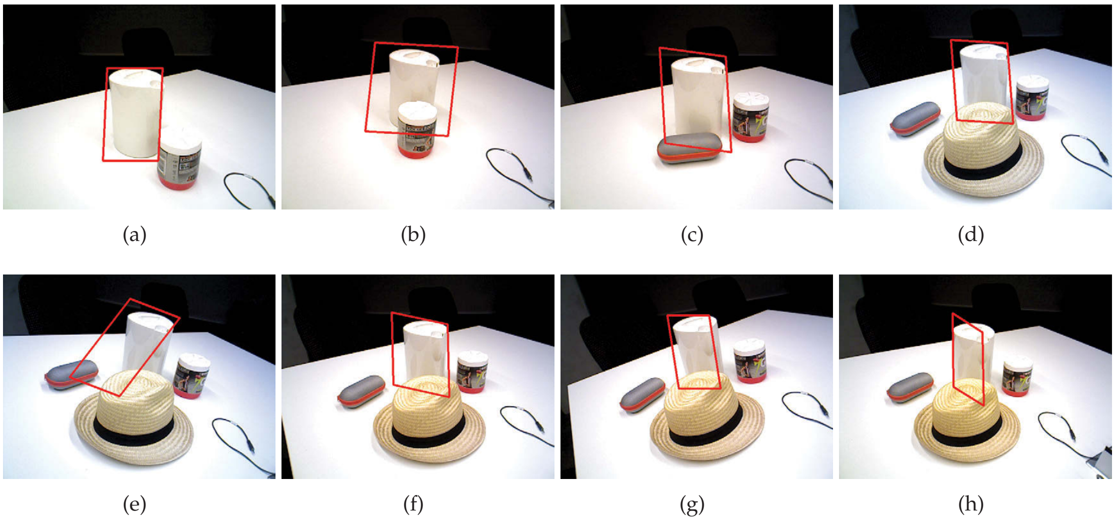


FIGURE 4.23: Object detection results corresponding to the object in Fig. 4.22.

with grayscale image to extract the keypoints using the principle similar to that of the Harris detector. An enhanced RGB-D descriptor is also proposed based on the previous proposed LOIND descriptor which significantly improves the matching performance. RISAS is demonstrated to be invariant to viewpoint, illumination, scale and rotation. RISAS detector is shown to enhance the performance of CSHOT and LOIND which are currently the best performing RGB-D descriptors. The performance of RISAS for object recognition under extreme illumination conditions is also demonstrated.

Here the author would like to highlight the connections and relations between this chapter and both Chapter. 3 and Chapter. 5. Due to the inefficiency of the current implementation of RISAS in Matlab, this thesis is unable to demonstrate its performance for active object detection and pose estimation in Chapter. 6 and Chapter. 7. However, as a point feature, RISAS can be adapted seamlessly into the work discussed in following chapters. Implementation of RISAS in C/C++ is one of the key activities planned for future work.

Chapter 5

Object Detection and Pose Estimation for Warehouse Automation

The first Amazon Picking Challenge (APC) was held alongside the 2015 IEEE International Conference on Robotics and Automation (ICRA) in Seattle, Washington, 26−27 May. The objective of the competition was to provide a challenging problem to the robotics research community that involves integrating the state-of-the-art object perception, motion and grasp planning algorithms to manipulate real-world items in industrial settings. To that end, APC posed a simplified version of the task that many humans face in warehouses all over the world, that is picking items from shelves and putting them into containers. In this case, the shelves were prototypical pods from Kiva Systems, and the picker had to be a fully autonomous robot.

In APC 2015, University of Technology, Sydney teamed up with Zhejiang University and Nanjiang Robotics Co. Ltd to form the team, Z.U.N., which was ranked No. 5 in the competition which was attended by 28 participating teams including the top universities around the world such as MIT and UC Berkeley. The team designed a dual-arm robot with a suction gripper. The perception system had two sets of the combination of an RGB-D sensor and a monocular camera to fully-cover the shelf.

This chapter demonstrates the effectiveness of the proposed perception system under realistic environment using the perception module which was designed for the APC 2015 as a case study.

5.1 Environment Setup for APC 2015

There were 24 items selected by the APC organiser which were commonly sold on `Amazon.com` and posed various degrees of difficulties in terms of both recognition, estimation and grasping, as shown in Fig. 5.1. Rigid and textured objects such as a box of straws and a box of pencils are examples of trivial cases. Some of the items are difficult because they are easy to be reshaped or damaged such as books and soft-packaged cat food. Transparent coverage items pose difficulties during the recognition and pose estimation step because of the reflective surfaces and the lack of depth information. Cheez-it box, even though being textured and rigid, needs to be twisted to be taken out from the bin as it is oversized, thus posing an extra challenge in grasping planning. Some other items such as the plush toys bring difficulties in the grasping step. One extreme case is the meshed pen holder which is difficult in both detection and grasping.

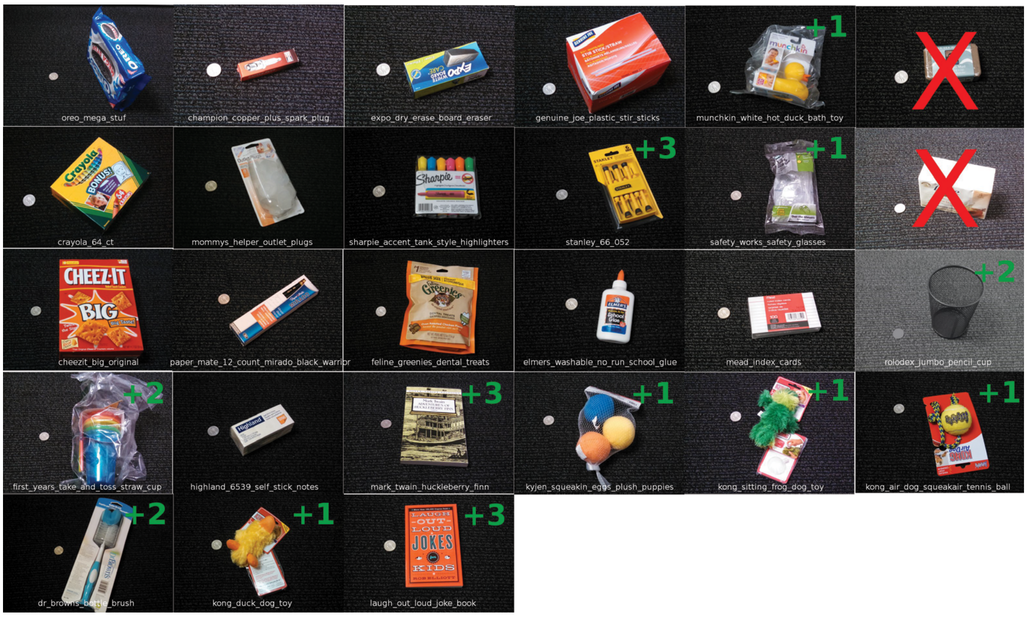

oreo_mega_stuf
champion_copper_plus_spark_plug
expo_dry_erase_board_eraser
genuine_joe_plastic_stir_sticks
+1
munchkin_white_hot_duck_bath_toy
crayola_64_ct
mommys_helper_outlet_plugs
sharpie_accent_tank_style_highlighters
+3
stanley_66_052
+1
safety_works_safety_glasses
cheezit_big_original
paper_mate_12_count_mirado_black_warrior
feline_greenies_dental_treats
elmers_washable_no_run_school_glue
mead_index_cards
+2
rolodex_jumbo_pencil_cup
+2
first_years_take_and_toss_straw_cup
highland_6539_self_stick_notes
+3
mark_twain_huckleberry_finn
+1
kyjen_squeakin_eggs_plush_puppies
+1
kong_sitting_frog_dog_toy
+1
kong_air_dog_squeakair_tennis_ball
+2
dr_browns_bottle_brush
+1
kong_duck_dog_toy
+3
laugh_out_loud_joke_book

FIGURE 5.1: Targeting objects for recognition and pose estimation in APC 2015

The 24 items were placed in a pod with 12 bins which is shown in Fig. 5.2. A bin may consist of a single item or multiple items which can be identical or different. Only one of these items is the target item which needs to be grasped safely from the bin. The score started from 10 for one successful pick-up and the score increased if additional items were distributed in the same bin together with the target object. Based on the specific characteristic of each item, some items were given 1 to 3 extra scores such as the plush toys and books. Damaging an item incurred a five-point penalty while picking the wrong item incurred a 12-point penalty. Each competitor had

20 minutes to pick as many of the 12 target items as possible. All the items with additional scores are also highlighted in Fig. 5.1 as well.

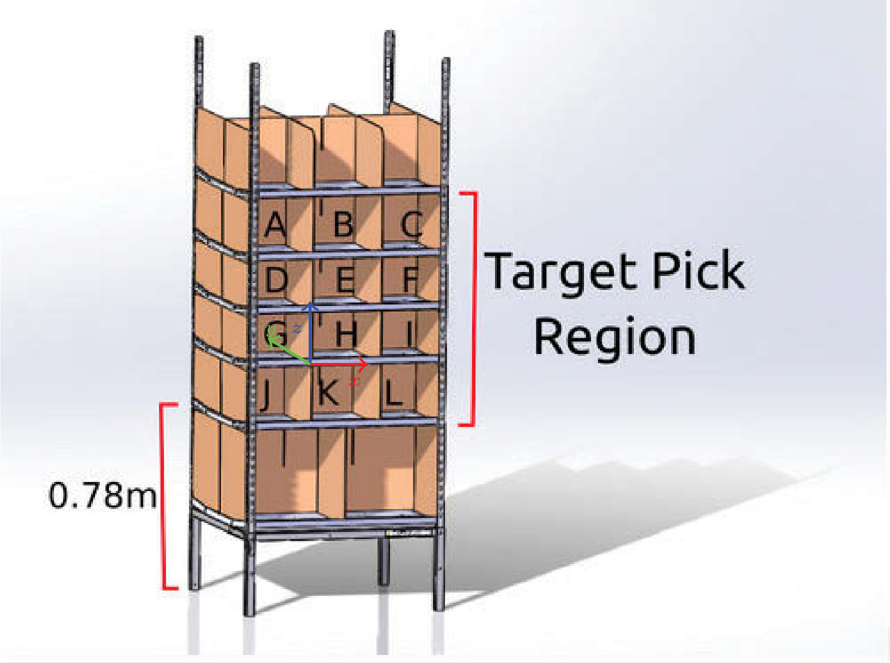


FIGURE 5.2: The shelf which contains the objects used in APC 2015.

The list below highlights some of the common difficulties faced in detecting and grasping the object reliably during the competition:

1. The walls and shelves are not equi-distributed. This introduces differences in the nominal size of the openings of each individual bin, with height ranging between 19 and 22cm, and width between 25 and 30 cm;
2. Each bin has a lip on the bottom and top edges, as shown in Fig. 5.3, which impedes exposing an object;
3. The lateral bins have a lip of the exterior edge, as shown in Fig. 5.3, which impedes exposing an object;
4. The metallic bottom of the structure produced bad reflections which proved to be an impediment for accurate estimation of the location of the shelf by model fitting to point cloud data.

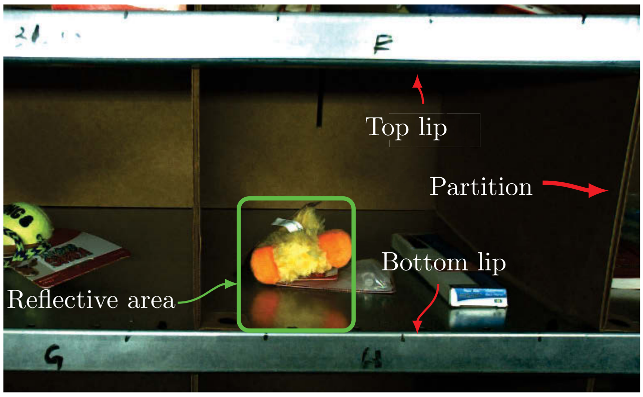


FIGURE 5.3: Difficulties in object detection and grasping.

5.2 Analysis of the Particularities of the Warehouse Pick-and-Place Problem

In APC 2015, compared to the work presented in Chapter. 3 and Chapter. 4, the environment set-up was significantly different. Some of the differences made the problem more difficult and some of them provided extra information which made the case easier. The key issues are listed below:

1. In both Chapter. 3 and Chapter. 4, the proposed algorithms are designed to recognise all target items from an unstructured environment seldom given prior knowledge. However, in APC 2015, prior knowledge of the structured environment and the objects were available.

   (a) The size of the shelf and the dimension of each bin are given;

   (b) The relative transformation between the "shelf" frame (highlighted in Fig. 5.2) and the robotic body frame can be pre-calibrated;

   (c) The target object in each bin is given and the other objects which are placed together with the target object in the same bin are also given;

2. In the previous two chapters, the full 6 DoFs relative pose $[\mathbf{R}, \mathbf{t}]$ of the detected object is required to be estimated. However, in APC 2015, successful grasping and manipulation of the object did not require the estimation of 6 DoFs pose. In fact, it was sufficient to detect the region of the object prior to planning the grasping in 3D space. Using the pre-calibrated transformation between the robot body frame and bin frame, it was only required to estimate the position of the target region w.r.t the bin frame.

Note: It is fair to say that the environment set-up in APC 2015 is more constrained and trivial compared to the proposed framework in Chapter. 3, thus making the object detection and pose estimation problem relatively easier.

3. In Chapter. 3 and Chapter. 4, most of the objects which can be handled by the proposed approaches are non-reflective and more importantly are rigid. In this perspective, the objects included in APC 2015 were much more challenging. Following list shows the major categories of items in the competition.

   (a) Soft-cover books and objects with a crushable package;

   (b) Objects in plastic and transparent package which shows invalid information on a depth sensor;

   (c) Non-rigid objects which can be easily deformed by external force;

   (d) Meshed objects;

   (e) Plush toys;

   (f) Very small objects;

   All the candidate objects have already been presented in Fig.5.1 and bonus points were available for difficult objects. In Fig.5.1, one coin with 17.91mm diameter is placed beside the object to approximately indicates the size of the object.

   Note: The targeting objects in APC 2015 is more difficult compared with the objects from the previous studies.

5.3 Hardware Design of the Robotic Platform

In this section, the hardware system of the robotic platform is presented including its main body, gripper and sensors that were attached to the the body to perceive the environment. As shown in Fig.5.4, the platform was designed to have dual arms that can be lifted up and down using the belt transportation system mounted on the back of the robot body. As the height of the shelf was about $1.7m$, the lift system was an essential part of the robot which allowed the arms to reach the items on different levels making it possible for sensors to observe the bins and the gripper to grasp the target items. The lift system allowed the robot body (arm and sensor) to reach 4 different heights corresponding to 4 rows on the shelf. Considering the length of the robotic arm, the width of the

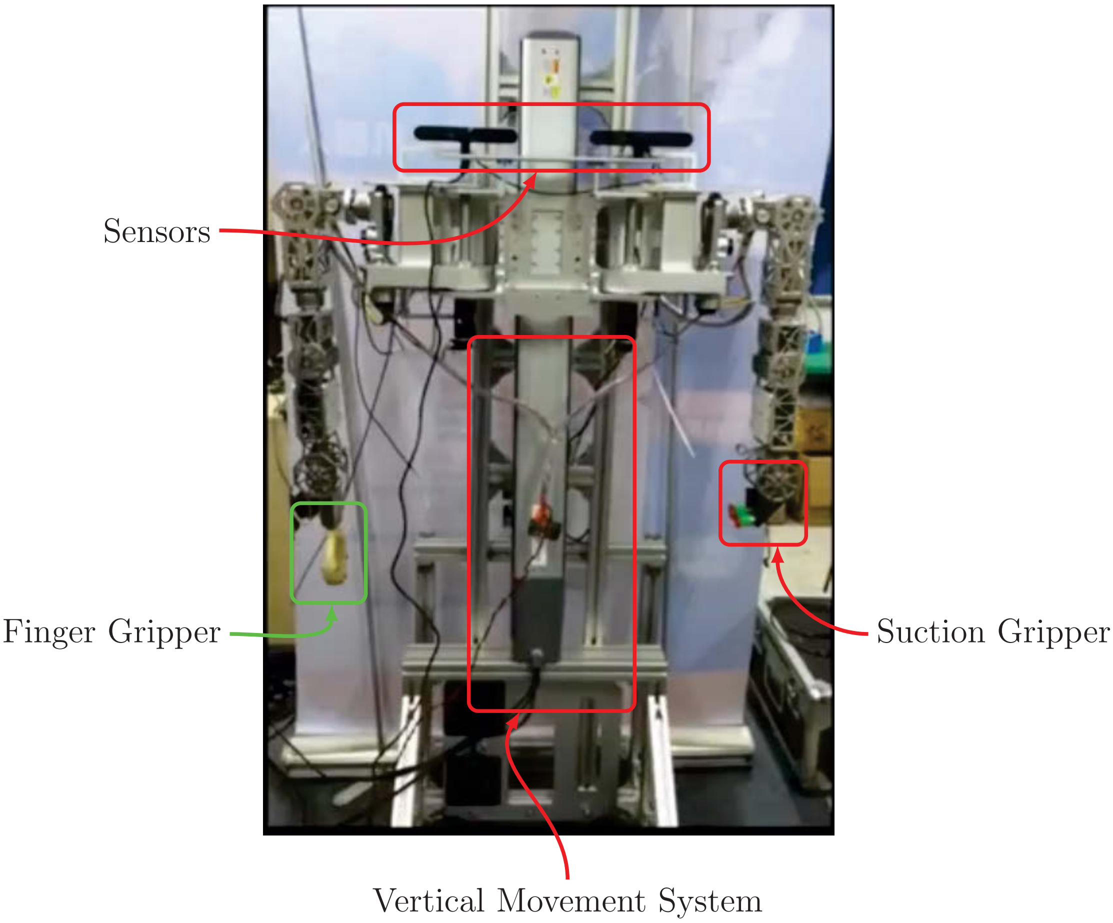


FIGURE 5.4: Robotic platform of APC 2015.

shelf and the Field-of-View (FoV) of the sensor (xtion RGB-D camera), 2 sets of the perception system were mounted on the platform. The one on the left hand side was responsible for the left side bins while the one on the right hand side was responsible for the right side bins. The bins in the middle column was covered by both sets. Each perception module is shown in the top red box in Fig. 5.4 and consists of:

1. An *Asus xtion* RGB-D camera which provides the depth information of the bin. This is the key sensor to provide information to estimate the pose;
2. A *pointgrey* high-resolution RGB camera which aims at recognising the targeting objects. The extrinsic parameter between the *xtion* and *pointgrey* are calibrated in advance;
3. An LED light which is used to control the illumination condition of the bin;

The following Fig. 5.5 illustrates the configuration of the sensor w.r.t to the shelf. Some basic and critical parameters of the robotic platform include:

1. The horizontal FoV of xtion sensor: This is 59° for the xtion sensor;
2. The distance between the object and the sensor to provide reliable depth information[1]: This distance was manged to be $0.5 \sim 0.6m$
3. The length of the arm is $0.68m$ and the length of the gripper is approximate $0.2m$: The robot platform cannot be placed too far from the shelf. In order to make both the sensor and the arm work properly, the distance between the sensor (robotic platform) and the shelf was set to be $0.6m$. Therefore, as shown in Fig. 5.5(a), it is not capable of covering the whole shelf. Fig. 5.5(b) provides a simplified view of the final configuration map, and the two sets of sensors as explained before.

In Fig. 5.4, the robotic platform is equipped with 2 different kinds of grippers: 1) a two-finger gripper on the right arm (left green box in Fig. 5.4) and 2) a vacuum gripper on the left arm (left green box in Fig. 5.4). The original idea was to grasp the rigid objects using the two-finger gripper and the objects with plastic cover using vacuum gripper. However, due to the size of the two-finger gripper and the reliability concerns, in the actual competition, both armed were equipped with vacuum grippers. During the experiments, except for the mesh pencil cup and one tiny object, the vacuum gripper was able to handle all the other objects well enough.

[1] Distance within $0.5m$ is the "blind-zone" for xtion sensor

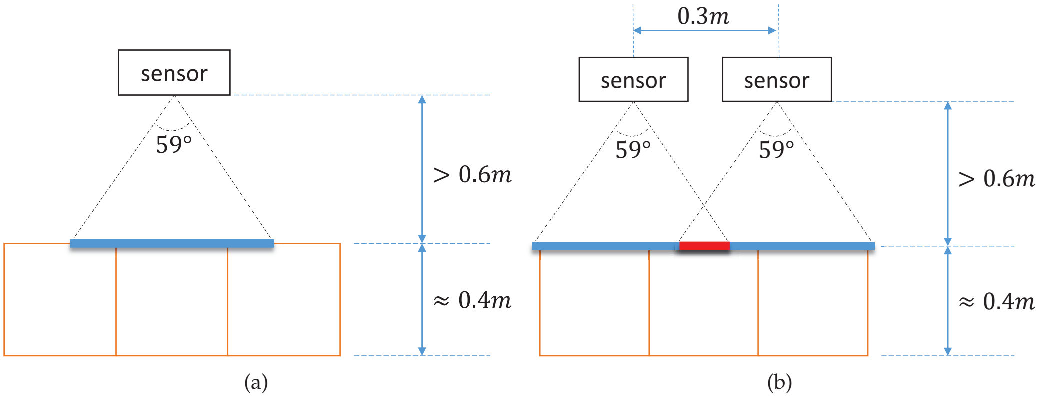


FIGURE 5.5: Sensor configuration for Amazon Picking Challenge.

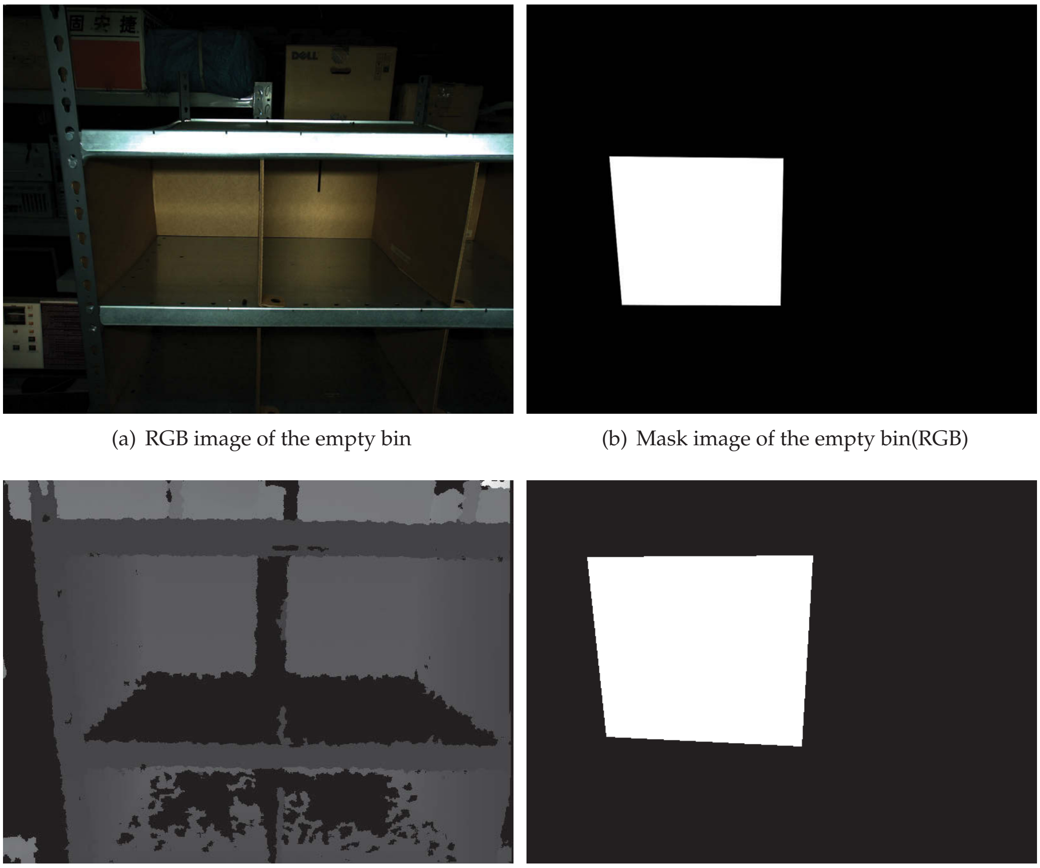
5.4 Perception Module

5.4.1 Pre-Processed Prior Knowledge

In order to provide more accurate and faster object detection and pose estimation performances, prior knowledge of the environment has been incorporated into the Perception Module. Besides pre-built object models and trained classifiers, the following information were also utilised:

1. RGB mask image of each bin;
2. RGB image of each empty bin;
3. Depth mask image of each bin;
4. Depth image of each empty bin;

A set of examples of images from bin *A* is presented in Fig. 5.6 below. Given the mask images (RGB and depth), the system is able to quickly identify the correct region which corresponds to the bin where the target object is located in. After that these reference empty images were subtracted to approximately identify the regions where these objects were placed. This allows to reduce the processing time substantially specially during the sliding window operation in the *Kernel Descriptor* recogniser and *EBlearn recogniser* . The details of the object detection using these two recognisers will be explained later in this chapter.



(a) RGB image of the empty bin

(b) Mask image of the empty bin(RGB)

(c) Depth image of the empty bin

(d) Mask image of the empty bin(Depth)

FIGURE 5.6: Pre-processed image for object detection.

5.4.2 Perception System Pipeline

The perception module that was designed for the APC is shown in Fig.5.7. The system accepts inputs from 2 sensor: a Xtion RGB-D camera with lower RGB resolution and a PointGrey high-resolution RGB camera. The extrinsic parameters between the depth sensor on Xtion and the PointGrey camera are calibrated[2].

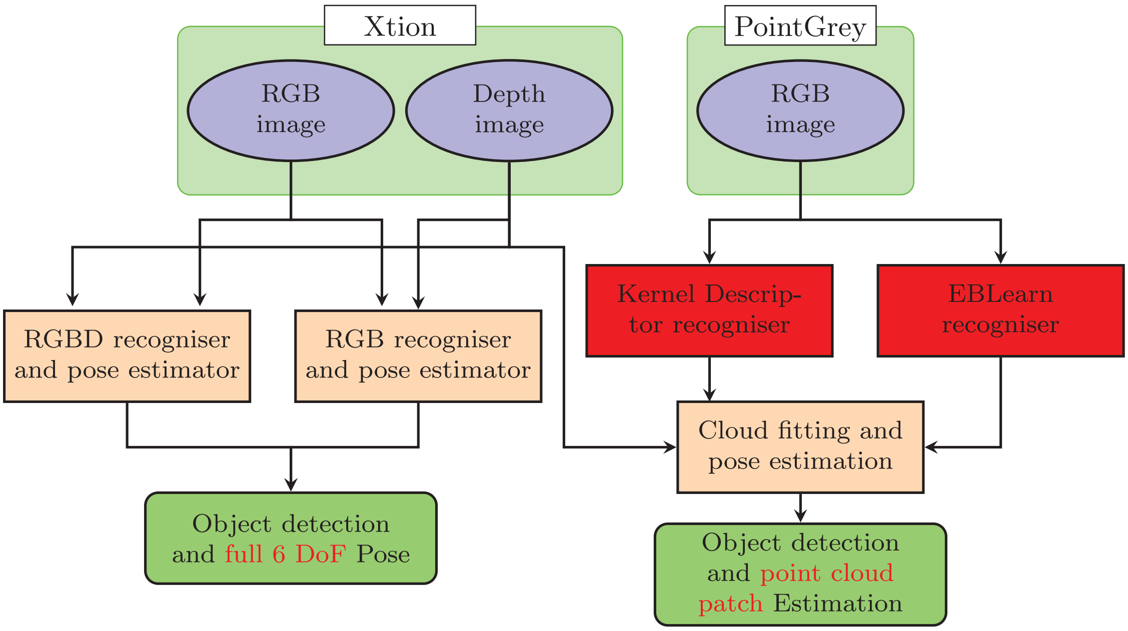

Xtion
RGB image
Depth image
PointGrey
RGB image
RGBD recogniser and pose estimator
RGB recogniser and pose estimator
Kernel Descriptor recogniser
EBLearn recogniser
Cloud fitting and pose estimation
Object detection and full 6 DoF Pose
Object detection and point cloud patch Estimation

FIGURE 5.7: Perception module for APC 2015.

As explained in section.5.1, the objects in APC 2015 are challenging due to inherent characteristics such as texture and deformation. The 24 objects were separated into two categories:

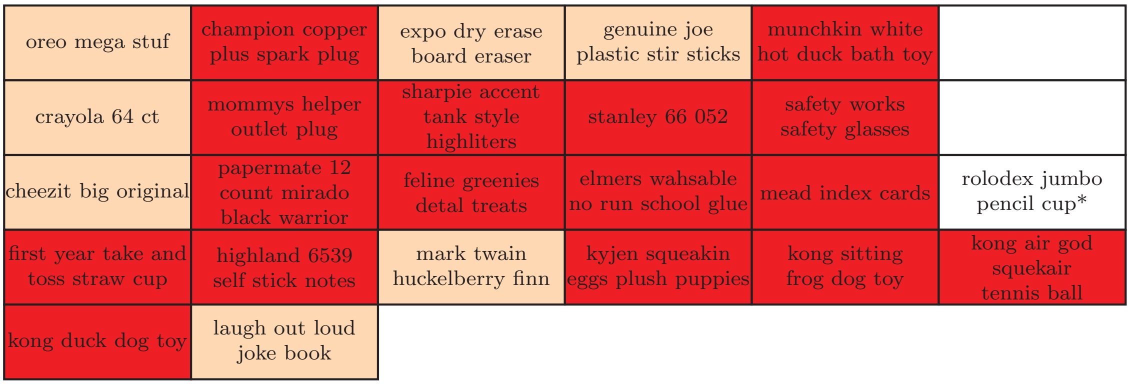
1. Textured objects with enough size;
2. Objects with plastic cover, small objects and deformable objects;

Fig. 5.8 shows the items of these tow categories where yellow items belong to category 1 and the red items belong to category 2.

For objects of category 1, the system is modified and implemented based on the proposed object recognition and pose estimation in Chapter. 3 . Some of the key modifications to cutomise the original system are:

1. Mask image: In order to extract the features only in the current bin, thus achieving better feature matching results, mask image is applied in feature extraction and selection step;

[2] Because the different resolution on depth sensor and PointGrey camera, there will be no guarantee that there is a depth value for each pixel on the RGB image provided by the PointGrey camera.



oreo mega stuf	champion copper plus spark plug	expo dry erase board eraser	genuine joe plastic stir sticks	munchkin white hot duck bath toy	
crayola 64 ct	mommys helper outlet plug	sharpie accent tank style highliters	stanley 66 052	safety works safety glasses	
cheezit big original	papermate 12 count mirado black warrior	feline greenies detal treats	elmers wahsable no run school glue	mead index cards	rolodex jumbo pencil cup*
first year take and toss straw cup	highland 6539 self stick notes	mark twain huckelberry finn	kyjen squeakin eggs plush puppies	kong sitting frog dog toy	kong air god squekair tennis ball
kong duck dog toy	laugh out loud joke book				

FIGURE 5.8: Selected recognition methods for each objects in APC 2015

2. Dynamic kd-tree construction: Rather than building a SIFT descriptor KD-tree of all 24 objects, the kd-tree is built using only the objects in the current bin in runtime;

3. Single object detection and pose estimation: In the proposed object detection system as well as in MOPED, multiple existing objects are detected. However, in this work, the key focus remains on the target object. Please note that the poses of the other objects are not considered in the perception module, and collision avoidance between the gripper and other objects is *not* considered;

For objects of category 2, the team adopted kernel descriptor[24] and EBLearn[132] based approaches. Different to RGB-Recogniser and RGBD-Recogniser, both KD-Recogniser and EBLearn-Recogniser can only provide the bounding-box of the detected objects. Therefore, in case of KD-Recogniser and EBLearn-Recogniser after the recognition step, another plane fitting and estimation step was incorporated. It is worth to mention some of the key strategies that we have adopted in implementing the object recognition software module.

1. Instead of using a sliding window based detector with KD-Recogniser and EBLearn-Recogniser, in order to capture target objects, image subtraction was adhered. The mask image, RGB and depth image of the empty bin were fully utilised to identify the image patch of the target objects.

2. In KD-Recogniser, the classifier for the target object A is trained using the images of object A and the background. Compared with training the classifier for object A w.r.t all other objects, this approach has the following benefits.

(a) Since there is only a limited number of objects in the bin at any given point of time, when using a classifier trained with all the objects, there is a high chance that the classifier will produce inaccurate classification results.

(b) Since there is no prior knowledge of the combination of objects in a given bin, it is required to either train the classifier online or provide all possible combinations of objects, and none of them is realistic.

3. All candidate image patches for a given bin are tested with the trained LIbSVM classifier and the one that shows gives the highest score is selected as the correct one.

4. For the EBLearn-Recogniser, models are trained for each object individually;

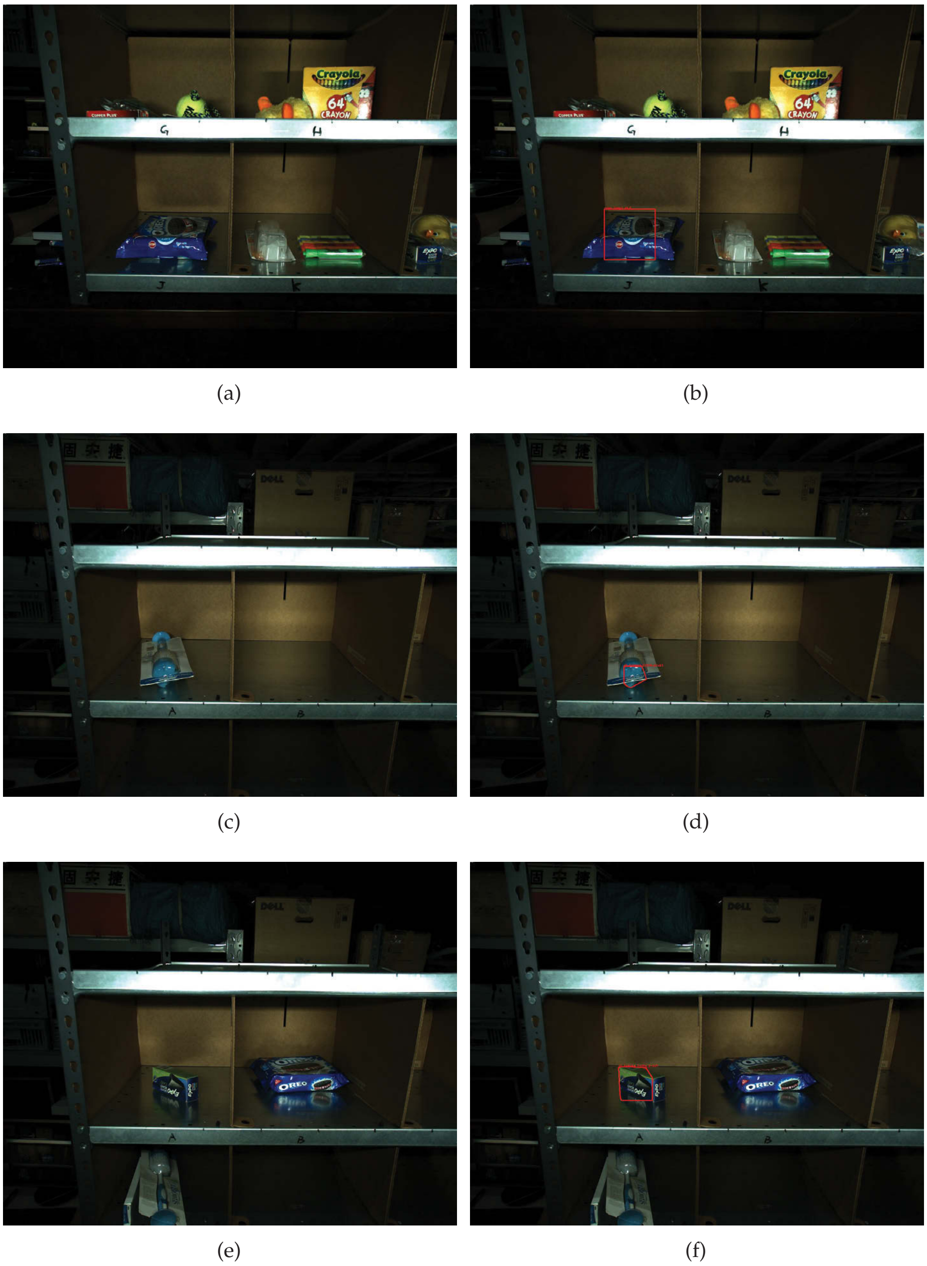
5. Without knowing the full 6 DoFs pose of the object, in order to grasp the object robustly using the vacuum gripper , the position of the surface patch where the vacuum gripper must be placed needs to be estimated.

The pose estimation for objects in category 1 is trivial since the 6-DoFs relative pose can be estimated from the RGB-D and RGB recogniser. For the challenging objects in category 2, by registering the RGB image with the corresponding depth image captured from xtion sensor, it is possible to obtain the point cloud of the detected objects. The proposed pipeline adopted surface fitting method available in PCL to determine the correct suction point.

5.5 Experimental Results

This section summaries some of the object detection results of the proposed system. Fig. 5.9 demonstrates the detection results when a single object is placed in the bin. Even though this single object case seems to be trivial, as shown in Fig. 5.9, it is difficult to identify the whole object given the imperfect depth information and reflective surfaces. For example, in Fig. 5.9(c) and Fig. 5.9(d), the cup brush object can be captured only in its bottom part. In Fig. 5.9(e) and Fig. 5.9(f), there exists a reversed reflection of the box due to the material of the surface.

Fig. 5.10 demonstrates more examples of multiple object detection results. The perception module designed and implemented for the APC is capable of providing reliable and robust object detection results even under occluded environment. In Fig. 5.10(a) and Fig. 5.10(b), parts of the duck toy and the box were under the shaded area, however the proposed framework is still capable of finding the object given limited observable information. The proposed framework can



(a) (b)

(c) (d)

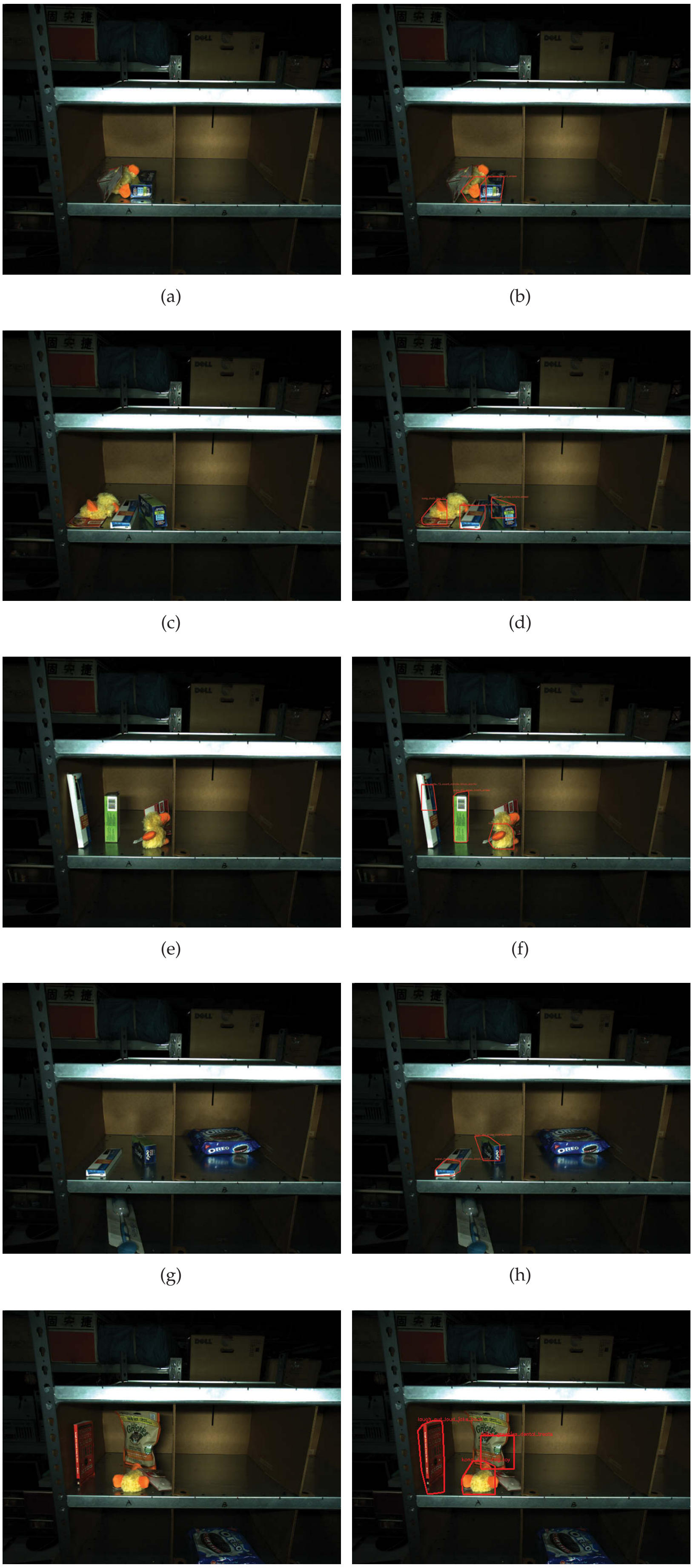
(e) (f)

FIGURE 5.9: Single object detection results.

also be used to detect target objects under the side-view where only very limited information is provided, e.g., the vertically placed books presented in Fig. 5.10(e), Fig. 5.10(f), Fig. 5.10(i) and Fig. 5.10(j).

5.6 Summary

This chapter has presented a robotic platform has been designed and implemented to use in the Amazon Picking Challenge 2015 competition. As a part of the system, a reliable and robust perception module that is suitable for the competition environment (similar to a typical warehouse environment) was designed and developed using the proposed work in Chapter. 3 and Chapter. 4. Additionally a Kernel Descriptor based recogniser and an EBlearn based recogniser were embedded into the pipeline to improve the efficiency and the accuracy. The proposed framework fully utilises the provided prior knowledge of the environment to achieve better detection results. Using the suction gripper on both arms, the robot can grasp the target objects from a specified bin given the detected bounding boxes. Using this setup, the Z-U-N team was able become one of the top-5 teams in the Amazon Picking Challenge 2015 competition. Unfortunately, the work in Chapter 4, the RISAS RGB-D feature is proposed after APC 2015, thus it was not utilised for less-textured object detection in this chapter.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

Part II

Active Object Detection and Pose Estimation

Chapter 6

Model-Driven Active Object Detection and Pose Estimation

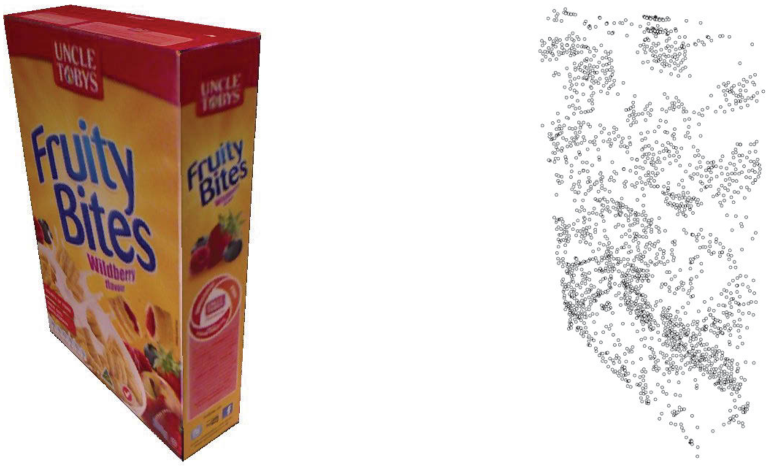
In **Part. 1**, from Chapter. 3 to Chapter. 5, the contributions in object detection and pose estimation under cluttered environments using a single observation have been presented. However, as explained in Chapter. 1, in challenging conditions such as under cluttered environments or ambiguity of the objects, a single observation is unable to provide sufficient information to identify the objects. Therefore, active object detection and pose estimation by manoeuvring robots in the environment is an effective approach towards addressing these issues.

In this chapter, a novel active object recognition and pose estimation system targeting household objects in everyday situations is presented. A sparse feature model, augmented with the characteristics of features when observed from different viewpoints is used for recognition and pose estimation while a dense point cloud model is used for storing geometry. This strategy makes it possible to accurately predict the expected information available during the Next-Best-View (NBV) planning process as both the visibility as well as the likelihood of feature matching can be considered simultaneously. In order mitigate difficulties with objects with similar appearances, an additional attribute is attached to each feature which denotes its uniqueness across all the objects in the collected dataset. Note that the shared features are assigned with a lower weighting value. The proposed strategy can identify the discriminative features of each object easily and guide the sensor to viewpoints which can differentiate the target objects better. The effectiveness of proposed active object detection and pose estimation framework using an RGB-D sensor is also demonstrated.

This chapter is structured as follows: Section. 6.1 explains how to build the models for the active object detection system while Section. 6.2 illustrates the proposed active object detection and pose estimation algorithm from the model perspective. An improved RGB-D object detection framework is briefly explained and the NBV search strategy is also presented. Section. 6.3 demonstrates the effectiveness of the proposed approach to real world problems and Section. 6.4 concludes this chapter and discusses the advantages and limitations of the proposed method.

6.1 Information Rich Object Modeling

6.1.1 Models for Active Object Recognition



(a) Dense point cloud

(b) Sparse feature cloud

FIGURE 6.1: Dense and sparse model of the object "Fruity Bites"

In this work, each object is represented using two models: *dense point cloud model* $\mathbf{M}^{\mathrm{d}}$ and *sparse feature cloud model* $\mathbf{M}^{\mathrm{s}}$. $\mathbf{M}^{\mathrm{d}}$ is a dense RGB point cloud which characterises the shape and texture information of the object, shown in Fig. 6.1(a). In $\mathbf{M}^{\mathrm{s}}$, each feature f_i^s consists of the 3D coordinate p_i^s in the object frame and a local feature descriptor $\mathbf{d}_i^s$ of point p_i^s such as SIFT is included in f_i^s. Both $\mathbf{M}^{\mathrm{d}}$ and $\mathbf{M}^{\mathrm{s}}$ can be easily built using off-the-shelf Simultaneous Localisation and Mapping toolboxes. In this work, an RGB-D sensor is used to construct these models by positioning around the object. The dense model $\mathbf{M}^{\mathrm{d}}$ is further refined using filtering and surface fitting manually. Feature coordinates in $\mathbf{M}^{\mathrm{s}}$ are generated from the optimised camera poses using consistent correspondences across multiple observations. Fig. 6.2 shows a snapshot of the object modelling step using the Turtlebot mounted with an RGB-D camera.

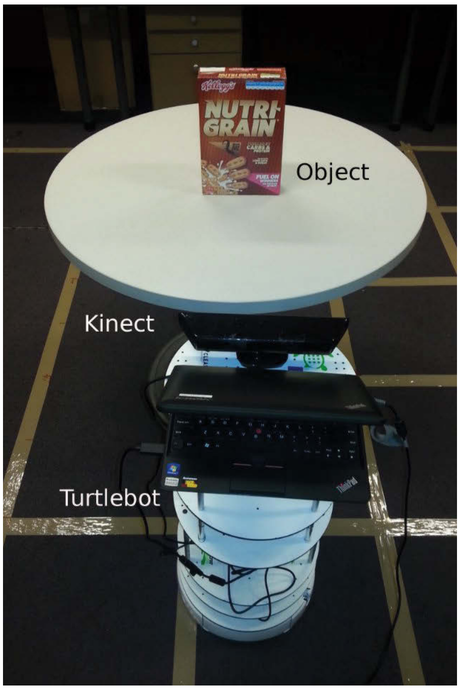
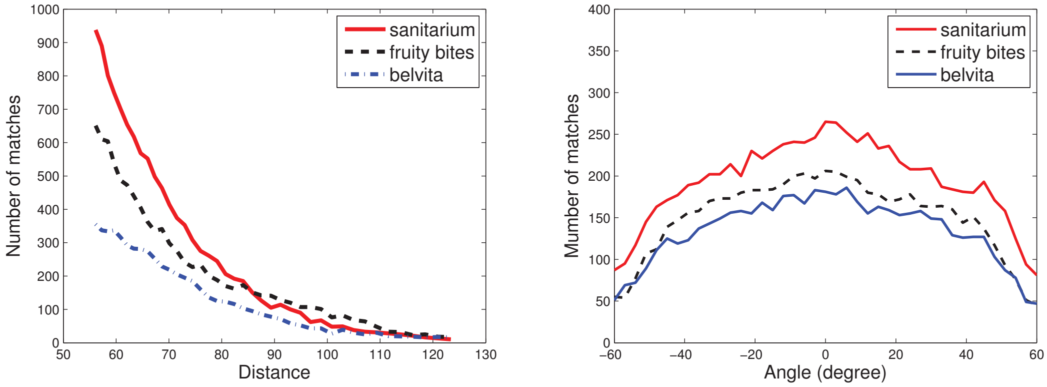


FIGURE 6.2: Object modelling using an RGB-D camera mounted on Turtlebot.

6.1.2 Information Rich Attributes for Better Prediction of Viewpoint Quality

One of the most important tasks that needs to be accomplished for active perception is the prediction of the quality of the information that is likely to be available from a selected viewpoint. Information quality depends on some visible features as well as the ability to associate the observed features to those included in the object model. Widely used feature descriptors such as SIFT and SURF, although designed to be robust to variations in scale and multiple deformations, fail to establish associations if such changes are too large. Fig. 6.3(a) illustrates the impact of the distance from the camera to the object for 3 objects ("Sanitarium", "Fruity Bites" and "Belvita") to the feature extraction and matching. Using Kinect sensor as an example, while more than 1000 features are captured for “Sanitarium” object when placed at a distance of 0.55 m from the camera, only 50 of these features are remaining to be detected and correctly matched at a distance of 1.2 m. It is seen from 6.3(a) that the number of correct matches decreases exponentially with the increase of the distance between features and the camera. It was also observed that the impact of the change in scale is a property of the region from which the feature is extracted and as such cannot be captured in one general formula.

Fig. 6.3(b) shows how the number of matches changes due to the variation of the angle between the surface normal and the camera axis at the range of 65 cm. Again it is seen that the number of features decreases while the angle between the surface normal and camera axis increases. It can be easily seen from this figure that change in the view angle also has a significant impact on the matching ability.



(a) Scale variation analysis

(b) Viewpoint variation analysis

FIGURE 6.3: The number of correct matches under different scale and viewpoint variations.

To address this issue, it is proposed to attach two attributes for each feature f_i^s: 1) *maximum observable distance* $d_{\max}$ and 2) *maximum observable angle* $\alpha_{\max} = g(d)$ which is a function of the distance d from the sensor to the feature. These extra attributes are used in prediction and depend on the type of the descriptor. $d_{\max}$ describes the maximum distance under which the feature can be reliably extracted and potentially correctly matched to the descriptor $\mathbf{d}_i^s$ in the model. $d_{\max}$ can be used to give a quantitative indication for the scale invariance of each feature. Similarly , $\alpha_{\max}$ denotes the upper bound of angle between the normal vector and the viewpoint, under which the associated feature can be repeated and correctly matched again. $\alpha_{\max}$ is a function of the distance between the camera and the feature. The function $g(d)$ allows larger $\alpha_{\max}$ in farther distance and smaller $\alpha_{\max}$ in closer distance.

The two parameters $d_{\max}$ and $\alpha_{\max}$ are set empirically. During object modelling, the local image patch of a feature is re-scaled into multiple levels and through extracting features and descriptors on scaled images, $d_{\max}$ is evaluated for each feature f_s in the model. $\alpha_{\max}$ is computed in a similar way by warping the local image patch using multiple levels of affine transformations. Section. 6.3 shows the impact of adding these two attributes in predicting the number of correct matches for the new viewpoint.

In following sections, it is demonstrated that both sparse model and dense model are essential for a reliable active object detection system. On one hand, The extracted features of the sparse model have a decisive influence on the performance of the object detection and the accuracy of pose estimation results. On another hand, the accurate prediction of observable features and the selection of NBV are heavily depended on the dense geometric models.

6.1.3 Feature Weighting using KD-Tree Structure

Motivation

Ambiguity caused by man-made texture is one of the major challenges in developing a robust object detection system. Similar textured objects can be observed frequently in daily life such as biscuit boxes with different flavours, as shown in Fig. 6.4. Deciding the NBV in active object detection using all available information (local features) may lead the sensor to an ambiguous viewpoint where similar objects cannot be reliably differentiated. Feature weighting and feature selection are well-known techniques used to address this issue by mining and discovering discriminative features. Feature weighting and selection techniques such as Term-Frequency – Inverse Document Frequency (tf-idf) and its variations have been widely acknowledged as a key step in image retrieval[112] and feature matching[145].

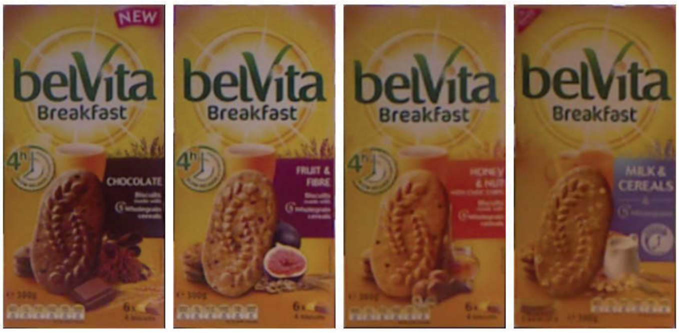


FIGURE 6.4: Belvita biscuit boxes of different favours

Even though feature weighting and selection can be critical in selecting the NBV for active object detection and pose estimation, its advantage and importance have not been fully recognised in the literature until recently. In Potthast et al.'s work[118], online feature selection became vital in making the decision between moving to the NBV or re-detecting the target object again using another type of feature under the same pose. In the framework presented in Section.6.2, the selection of NBV is closely related to the quality of expected correspondences under a selected viewpoint where the quality is characterised by not only the number of matches but also the uniqueness of the matched features. Since the feature matching is realised through the nearest neighbour searching in kd-tree, this study aims at adding weight to each feature which indicates its distinctiveness with respect to the object it belongs to.

Method

Kd-tree is a popular data structure for organising k dimensional data. At each non-leaf node, the k dimension space is separated into two half-spaces by a hyperplane. With a kd-tree which consists of N leaves, given a query feature, it requires $\log_2(N)$ comparisons to identify the nearest single leaf node. Kd-tree is regarded as an important nearest neighbour searching technique for various applications[102][103].

In the object recognition system presented in[31] which uses local features such as SIFT, feature matching using nearest neighbour searching algorithm is a critical step to achieve reliable object detection performance. For this step the kd-tree approach is frequently applied. However, when facing multiple ambiguous objects, there are similar features from different objects and each feature is attached to one leaf of the tree. When noisy feature descriptor is captured and queried into the kd-tree, it can be matched to the feature from incorrect objects thus leading to inaccurate detection results. Therefore, from practical viewpoint there is a necessity to add a weighting parameter that captures the uniqueness of each feature.

Algorithm 2: Feature weighting in kd-tree

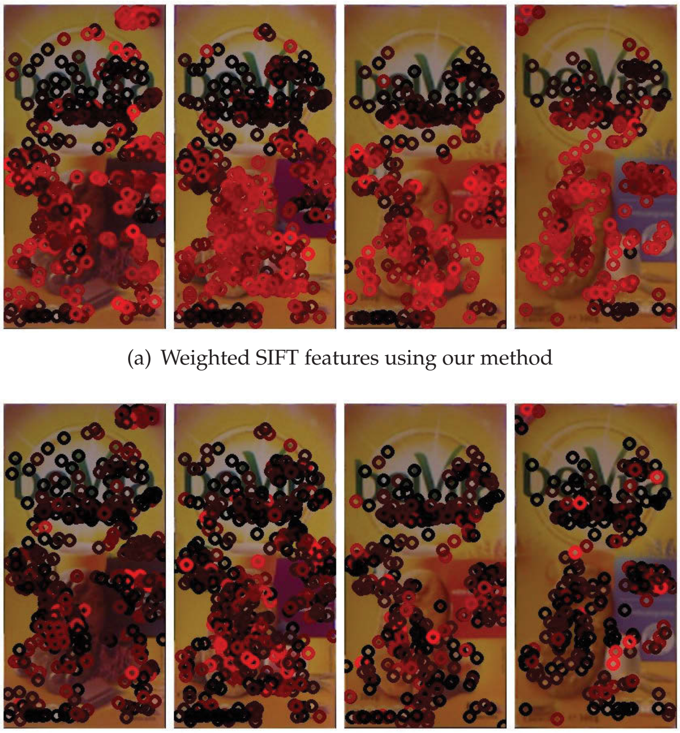
```
Input: All objects $\mathbf{O}$ and features $\mathbf{f}$ where $\mathbf{f}_i$ denotes features from object $O_i$;
Output: Weighted kd-tree $\mathbb{T}_w$;
$\mathbb{T} = \mathbf{BuildTree}(\mathbf{f})$// build kd-tree using all features;
foreach $f$ in $\mathbf{f}$ do
    $\mathbf{f}_{nn} = \mathbf{NNSearch}\,(\mathbb{T}, f)$;
    foreach $f_j$ in $\mathbf{f}_{nn}$ do
        if $\mathbf{Label}(f)\; != \mathbf{Label}(f_j)$ then
            $\omega_f = |f, f_j|$ // distance between $f$ and $f_j$ Assign weight $\omega_f$ to feature $f$ in tree $\mathbb{T}$
        end
    end
end

// Functions:
// $\mathbb{T} = \mathbf{BuildTree}(\mathbf{f})$ : build kd-tree given features $\mathbf{f}$;
// $\mathbf{f}_{nn} = \mathbf{NNSearch}\,(\mathbb{T}, f)$: search nearest neighbours using tree $\mathbb{T}$ and query feature $f$;
```

In the feature matching step in Section. 3.2.2, ratio test is applied in deciding whether the correspondence is trustworthy and a fixed value γ is set on the (d_1^{nn}/d_2^{nn}) where d_i^{nn} is the distance between the input feature and its i^{th} nearest neighbour in the kd-tree. In this case, assuming the noise of the extracted features follow the same distribution during observation if the matched feature from model M^s is closer to features from another object, it will more easily lead to an incorrect correspondence.

In this work, a brute-force feature weighting scheme is proposed in the kd-tree structure using the Euclidean distance between the input feature and its nearest neighbour from other objects, as shown in Algorithm.2. Despite its simplicity, this technique achieves reasonable results in finding the similar SIFT features for each object and by adding additional weight to each feature, the planned trajectory can successfully differentiate similar objects presented (detailed in Section.6.3). Other feature weighting schemes such as TF-IDF have also been adopted into our framework for comparison purposes. Section 6.1.3 presents the comparative results and shows the effectiveness of the proposed approach.

Experiments and Results



(a) Weighted SIFT features using our method

(b) Weighted SIFT features using TF-IDF after K-Means clustering

FIGURE 6.5: Feature weighting especially for similar object

To validate the feature weighting method, in this Section, the focus has specifically been on 4 similar objects (shown in Fig.6.4) to compute the uniqueness of each feature on 3D object models.

As presented in Fig.6.5(a), the red circles denote the unique feature and the dark circles denote the features which are similar to multiple objects. For better clarification, only the weighted features in the front face of the Belvita boxes are shown. Fig.6.5(b) shows the feature weighting results by adopting TF-IDF which is calculated in grouped features after K-Means clustering for all the features from every object. Even with different types of clustering methods and TF-IDF variations, the feature weighting results are still not satisfactory compared with the proposed approach.

Discussion

In this Section, a brute-force weighting method was proposed by querying every feature into the kd-tree. It helps in identifying the discriminative features and assigning reasonable weights among similar and even general objects, as described in Section.6.1.3. However, there are other possible methods to speed up the discriminative feature weighting process such as exploiting the tree structure in each layer. It is not surprising that instead of querying features one-by-one, it is more efficient to visit multiple nodes in the tree and analyse the Euclidean distances among leaves on each node simultaneously. Also, note that kd-tree may not be the best way to finding the nearest neighbours of image feature descriptor [84]. Experiment results from Kumar et al. indicated that Vantage Point tree(vp-tree) is a better option compared with kd-tree. We will further validate the vp-tree for feature selection in future research work.

The proposed method shows superior results compared with other widely-acknowledged methods due to the following two main reasons.

1. Most of the feature weighting methods in image/text retrieval require a Bag-of-Words representation of the features thus a clustering step is inevitable. However, the performance of clustering high-dimension data still cannot be reliable[137];

2. Most popular feature weighting and selection methods do not address the key problem outlined in this thesis. TF-IDF, for example, requires the features to appear frequently on the target objects (TF) and has less dispersion across other objects (IDF) at the same time. However, the first constraint is not applicable to this problem;

The proposed approach shares the similar principle with the *near-miss* in RELIEF feature weighting method described in[80]. The key difference is that in RELIEF weights are designed on different dimensions of the feature. The influence of using weighted features on active object detection is demonstrated in Section.6.3.

6.2 Active Object Recognition and Pose Estimation System

6.2.1 Object Recognition and Pose Estimation Using a Single Viewpoint

This section presents the framework for obtaining initial hypotheses as to the objects present in a cluttered scene and their relative poses from information acquired from a single observation. The framework presented in this section, shown in Fig. 6.6, is based on the proposed algorithm in Section. 3.2. The dashed green boxes denote the newly introduced steps which will be detailed in this section.

RGB-D Segmentation

In Section. 3.2, mean shift clustering method is adopted to cluster feature correspondences after matching. However, in this chapter, the order of the steps is slightly different. First, the RGB-D point cloud is obtained from the depth image and the RGB image. Then, the RGB-D point cloud of the whole environment is segmented into multiple groups using the pre-processing method described in Richtsfeld [122]. A computationally light-weight version of [122] is implemented on the robotic platform. Two different models, planes and NURBS (Non-Uniform Rational B-Splines), which describe the geometry of the object are used for fitting and segmenting the surface patches in the point cloud. Given that most man-made household objects have planar surfaces or curved shape that can be easily described using NURBS, these two models can capture informative patches in the input point cloud which are most likely to contain objects. Examples of RGB-D segmentation results are shown in Fig. 6.7. In this framework, each cluster is not limited to contain one object only, by *pose refinement and combination* step, multiple objects are allowed to be combined into one cluster and one object can be separated into multiple clusters as well, as discussed in Section. 3.2.6. Compared with 3D mean shift clustering method, this RGB-D segmentation takes much time as presented in Table. 6.1. However, as demonstrated in Fig. 6.7 where the top row shows the input RGB images and the bottom row presents the segmentation results (note that the segmented patches are shown in different colours), the clustering result from the segmentation method is more consistent.

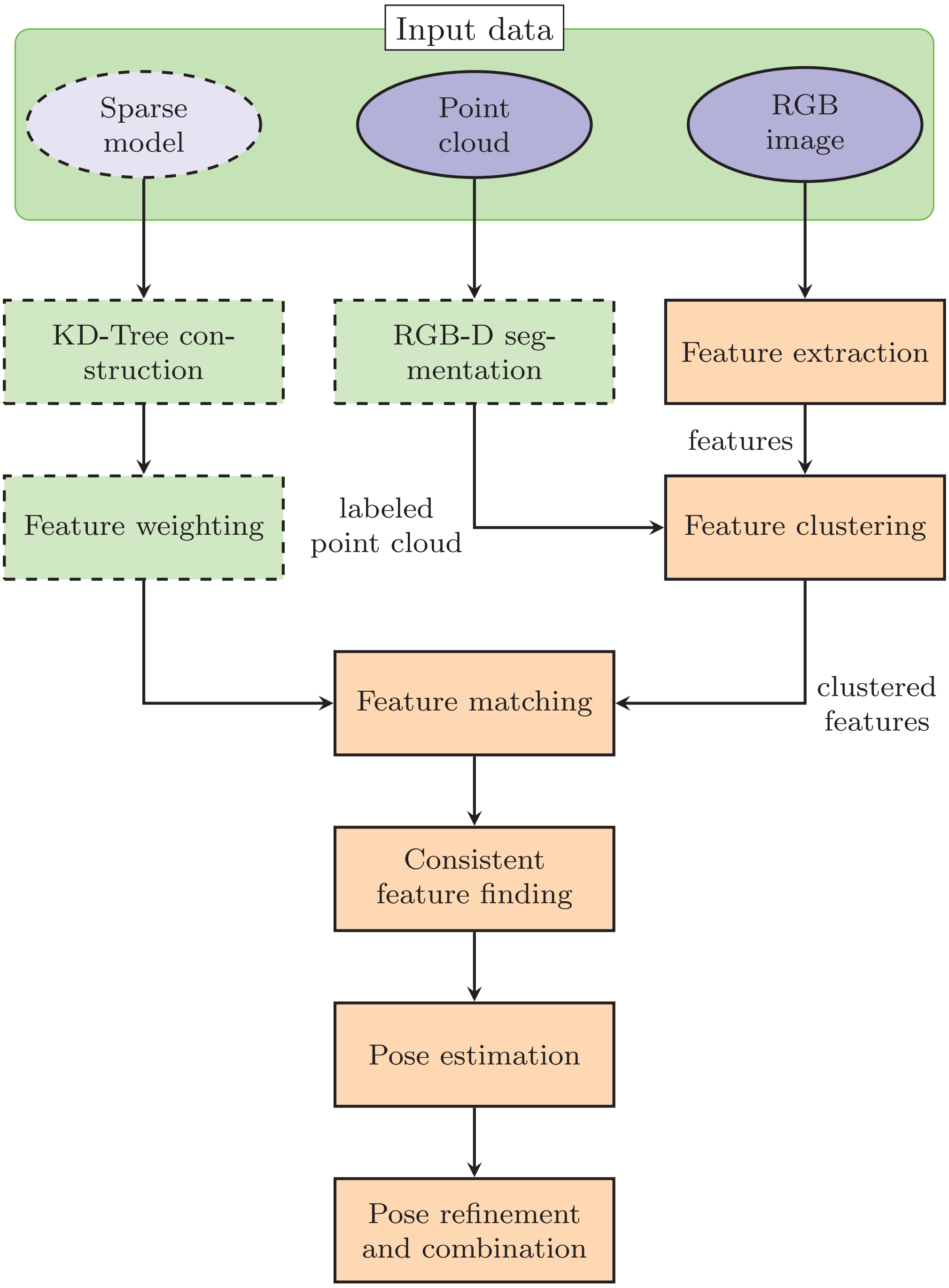


FIGURE 6.6: The system framework of object recognition and pose estimation using single RGB-D observation

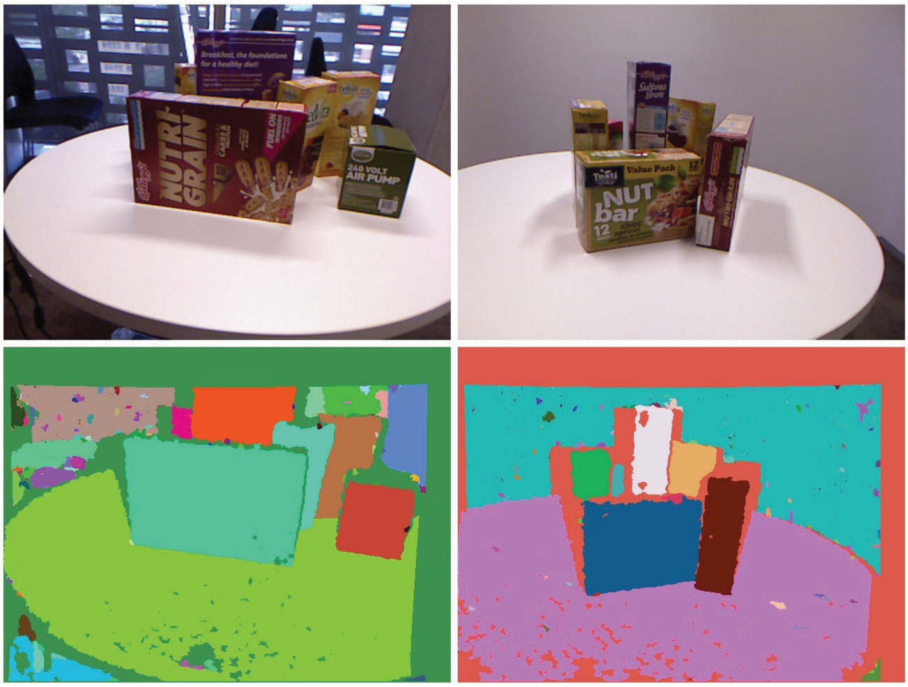


FIGURE 6.7: RGB-D segmentation results from cluttered environment.

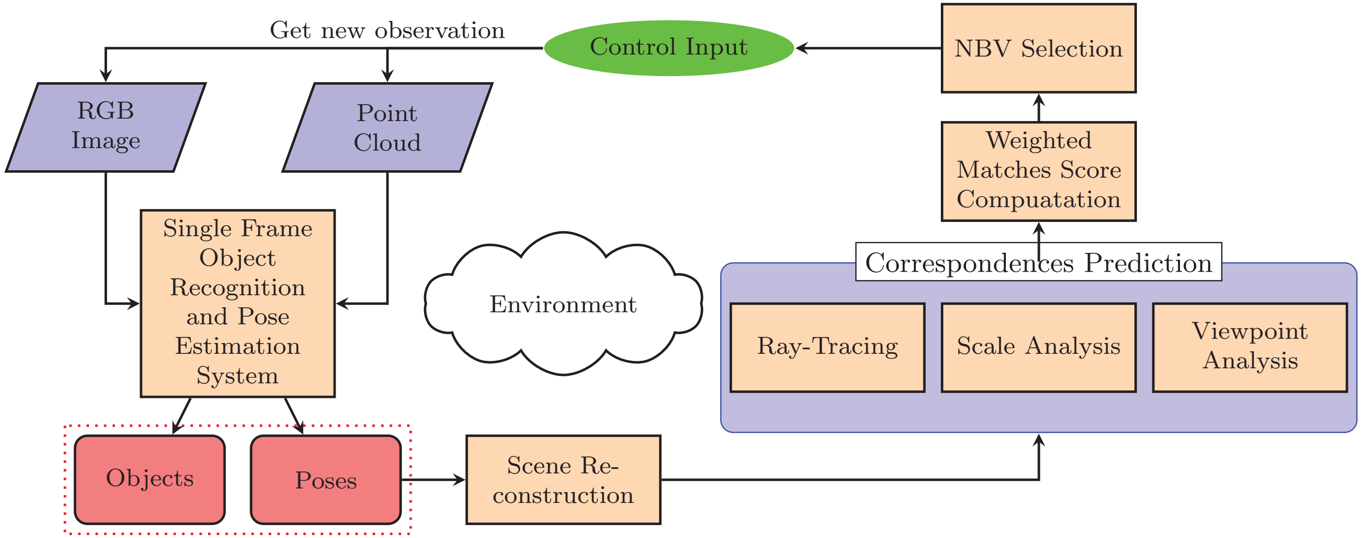


FIGURE 6.8: Framework of the proposed active object recognition and pose estimation system

6.2.2 Next-Best-View Selection using Information Rich Model

The virtual representation of the scene that encapsulates the current belief about the objects that are present in the scene and their poses can now be used to evaluate the quality of the information that can be expected if the robot moves to a new viewpoint. Therefore, it is possible to evaluate the utility of nearby locations so that the robot can then be moved to the Next-Best-View. This process can be repeated until some termination criteria such as stable object recognition and pose estimation, over a set number of moves is reached.

The overall flowchart of the active object recognition system is shown in Fig. 6.8. It begins with the virtual representation of the scene generated based on the information captured at the first robot pose. The three-step strategy is then used to predict the observable features which can be matched correctly from a new viewpoint.

1. **Raycasting**: In this step, all the features that are not visible from the new viewpoint due to occlusion by itself or other objects are rejected using raycasting by an octree structure. The world space is voxelised and given a starting point (observation position) and destination point (feature), all the voxels intersected by this straight-line are obtained. Therefore, it is possible to find out whether any features exist in the intersected voxels. This step can be completed in a few milliseconds. The detailed timing performance is presented in Section.6.3.2.

2. **Scale analysis**: Even though a feature point is observable, as discussed in section 6.1.2 there is no guarantee that this feature is detected and matched correctly. In the scale filtering step, a feature is removed from the candidate set if the distance between the sensor and the feature point, d_f, is larger than d_{max}, that corresponds to the feature under consideration.

3. **Viewpoint analysis**: Another key factor that influences the repeatability of feature matching is the viewpoint variation. Under the same distance, a feature can be re-detected and matched correctly under a limited range, as shown in 6.3(b). In the experiments presented in this section, a consistent threshold α_{max} is set for all the features in the models. The vector from the feature to the sensor is denoted as $\mathbf{v}_f$ and the normal vector of the local patch is denoted as $\mathbf{v}_n$. If the angle between $\mathbf{v}_f$ and $\mathbf{v}_n$ is larger than α_{max}, it is assumed that the feature cannot be correctly matched.

Using the above three steps, the number of potentially correct matches n_m under a given viewpoint can be predicted and the quality of a new viewpoint can be evaluated. In this chapter,

a simple yet effective greedy search approach is adopted in finding the NBV in the neighbourhood region of the current pose consecutively. During the active viewpoint planning, the observation history of each features until step k is recorded in *observation matrix* $\mathbf{O}_k \in \mathbb{R}^{n_f \times k}$ which includes under which poses each feature is detected where n_f is the sum of features on each object. Without considering the feature weight as discussed in Section. 6.1.3, entry $o_{i,j}$ is equal to 1 if feature i is detected on step j. Otherwise, considering the feature weight, $o_{i,j}$ is set as the weight of feature i. From $\mathbf{O}_k$, two values are generated which summarise the feature observation history:
1) n_{obser} which counts the number of features which have been observed before 2) n_{times} which sums all the non-zero entries in $\mathbf{O}_k$.

During the planning phase, for all candidate viewpoints in the neighbourhood of the current pose k, the observation matrix $\hat{\mathbf{O}}_{k+1}$ is predicted for each candidate and $\bar{n}_{\text{obser}}$ and $\bar{n}_{\text{times}}$ are further computed. The greedy search and selection criterion for this step is as follows.

1. If there exists $\bar{n}_{\text{obser}}$ larger than current n_{obser}, select the viewpoint which generates the largest $\bar{n}_{\text{obser}}$ thus observing more number of previously unseen features;

2. If $\bar{n}_{\text{obser}}$ equals to n_{obser} for all candidate viewpoints, $\bar{n}_{\text{times}}$ is used as the criterion and the viewpoint which generates the maximum $\bar{n}_{\text{times}}$;

3. If all $\bar{n}_{\text{obser}}$ and $\bar{n}_{\text{times}}$ happen to be the same for both n_{obser} and n_{times}, the robot follows the previous exploring direction or moves to a random direction if it is in the first step;

This criterion always enables the camera to move to a new position to acquire new information. The path planning in each step follows the logic of information gain which is widely used in active object recognition and autonomous object modeling. Given that the quality of information gathered from a given viewpoint is available, work proposed in this section can be adapted for use with more sophisticated trajectory planners. During the planning phase, it should be noted that there is no fusion for object detection and pose estimation from multiple observations, and the robot only trusts the detection results from the viewpoint which observes the largest number of features on the target object.

6.3 Experiments and Discussion

In this section, an off-line implementation of the proposed active object detection and pose estimation framework using a Microsoft Kinect sensor is presented. The key objective here is to validate and evaluate the performance of the proposed method. Note that this method can be easily extended to online active object detection with an external positioning system [82] or an accurate motion control system. Active object detection and poses estimation is tested in both 2D and 3D environments in this section as detailed in Section. 6.3.1 and Section. 6.3.2. A significant difference is that the 2D scenario only contains unique and different objects while the 3D environment contains both different and similar objects. The experimental results verify the effectiveness of the proposed method.

6.3.1 Case Study 1: NBV Selection in 2D environments

Experimental Set-up

In these experiments, a Turtlebot with a Microsoft Kinect mounted on the top of it (shown in Fig. 6.9) was used. The motion of the RGB-D sensor was constrained to 2D space. Eight *different* objects were placed on the table with different orientations. Due to the presence of occlusions, all objects cannot be observed at the same time from one viewpoint. Thus the robot needs to move to recognise all the objects and to estimate their poses.

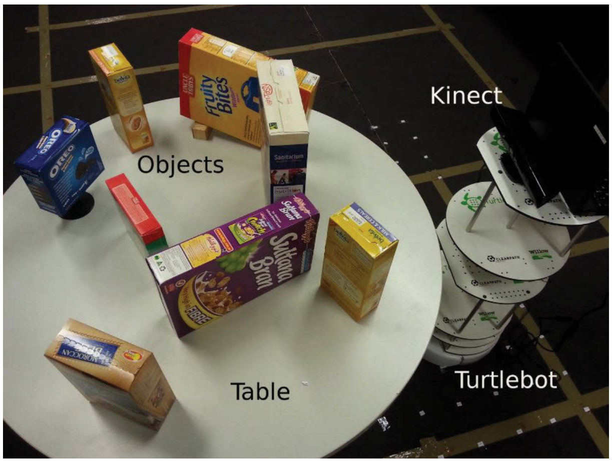


FIGURE 6.9: Active object recognition and pose estimation using Turtlebot

In the experiments reported in this subsection, the environment was divided into multiple cells, and the robot was manually placed in each cell as dictated by the planning algorithms. For simplicity, it is assumed that the robot orientation will be such that the camera will face towards the objects from each of the grid cells. The locations in which the robot can be positioned are shown in Fig. 6.10. 180 RGB-D images were collected by placing the robot in these locations so that the algorithm can be evaluated off-line.

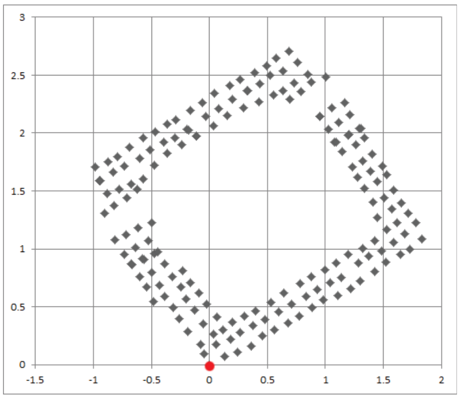
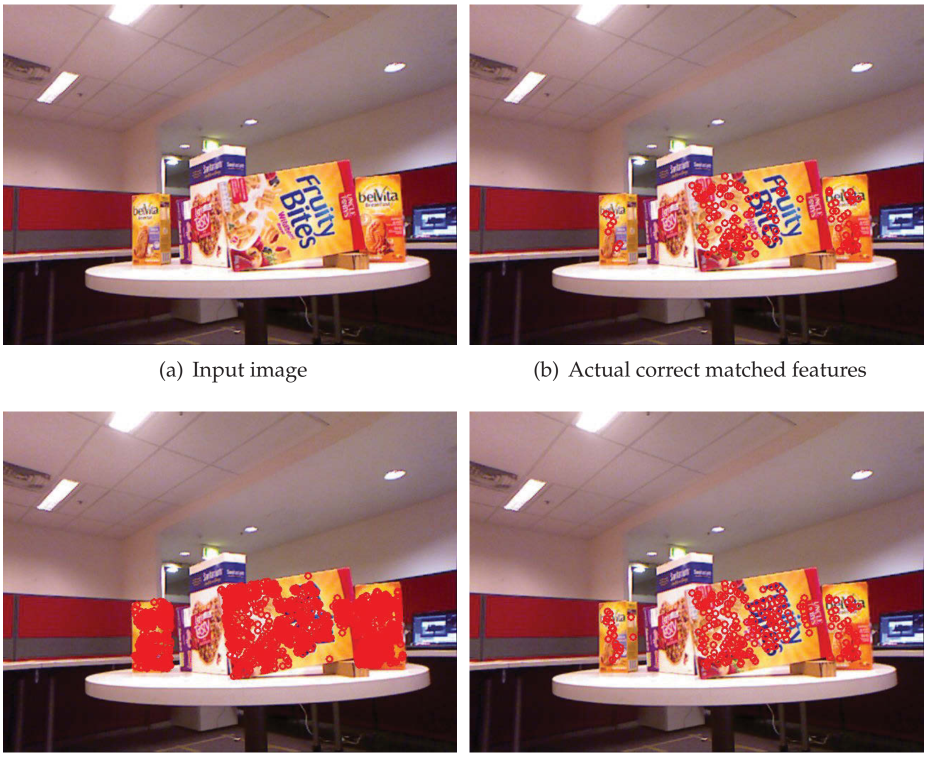


FIGURE 6.10: Optimised localisation using Parallax BA

Results and Discussion



(a) Input image

(b) Actual correct matched features

(c) Predicted matches without using $d_{\max}$ and $\alpha_{\max}$

(d) Predicted matches using $d_{\max}$ and $\alpha_{\max}$

FIGURE 6.11: Prediction of potential matches in next frame

1. **Prediction of Possible Matches:** As explained in Section. 6.1.2, a key advantage of the proposed algorithm is to provide more accurate prediction of the observable features. Under this scenario, Fig.6.11 shows the differences between prediction results with and without using the additional information $d_{\max}$ and $\alpha_{\max}$stored in the object model. Fig.6.11(a) is the

actual acquired image in frame 2 and Fig.6.11(b) shows the matched features when the robot moves to the 2nd frame. Fig. 6.11(c) shows the prediction outcome when $d_{\max}$ and $\alpha_{\max}$ are not used. It can be seen that the predicted matches using the proposed strategy that uses this additional information shown in Fig. 6.11(d) is much closer to real scene.

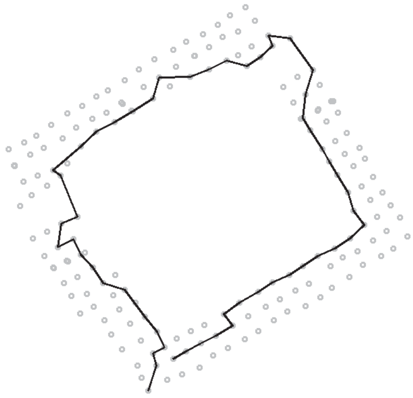


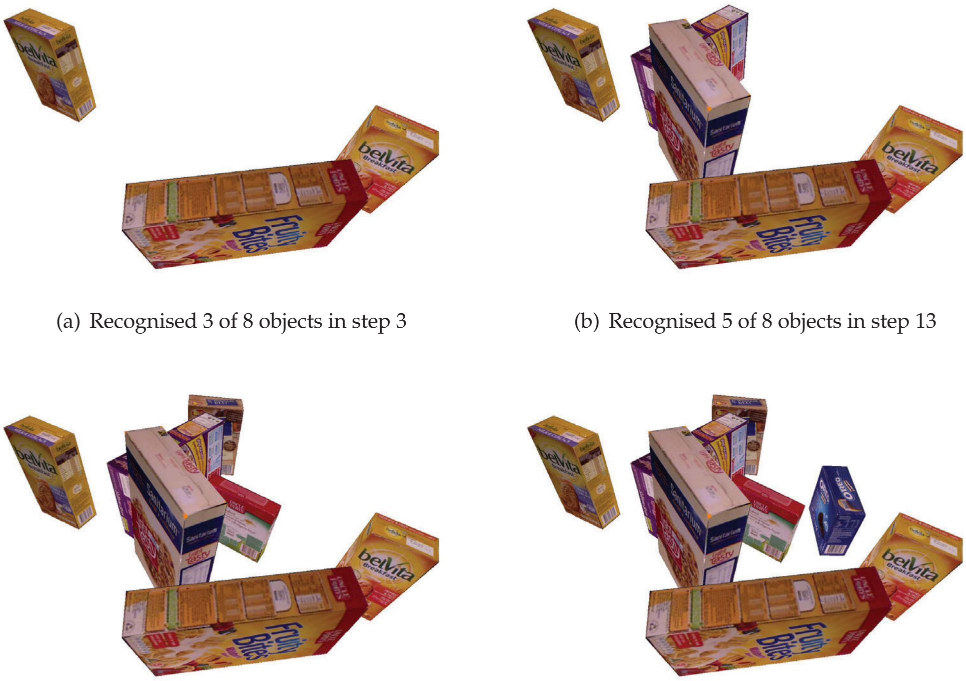
FIGURE 6.12: Planned trajectory for active object recognition and pose estimation

2. **Planned Path and Reconstructed Scenario**: Using the collected data at every pose shown in Fig. 6.10, a path is generated which can cover the whole space on the table. The planned path is shown in Fig. 6.12. When the robot moves along this path, the objects can be recognised one-by-one and finally all the objects on the table can be covered. Parts of the reconstructed scene during the motion are shown in Fig. 6.13. In Fig. 6.13(d), all the objects are recognised with accurate pose estimation results.

6.3.2 Case Study 2: NBV Selection in 3D environments

Experimental Set-up

Similar to Section. 6.3.1, the observation data is pre-collected, and the ground-truth poses are also calculated using off-the-shelf RGB-D SLAM algorithm as shown in Fig. 6.15. The algorithm starts from a selected pose in Fig. 6.15 and plans the trajectory for detecting target objects and estimating their poses autonomously. Compared with Section. 6.3.1, there are two obvious differences: 1) the poses of the sensors are distributed in 3D spaces 2) there are similar appearance objects placed in the environment.



(a) Recognised 3 of 8 objects in step 3

(b) Recognised 5 of 8 objects in step 13

(c) Recognised 7 of 8 objects in step 47

(d) Recognised 8 of 8 objects in step 52

FIGURE 6.13: Object recognition and pose estimation results in different steps of the path

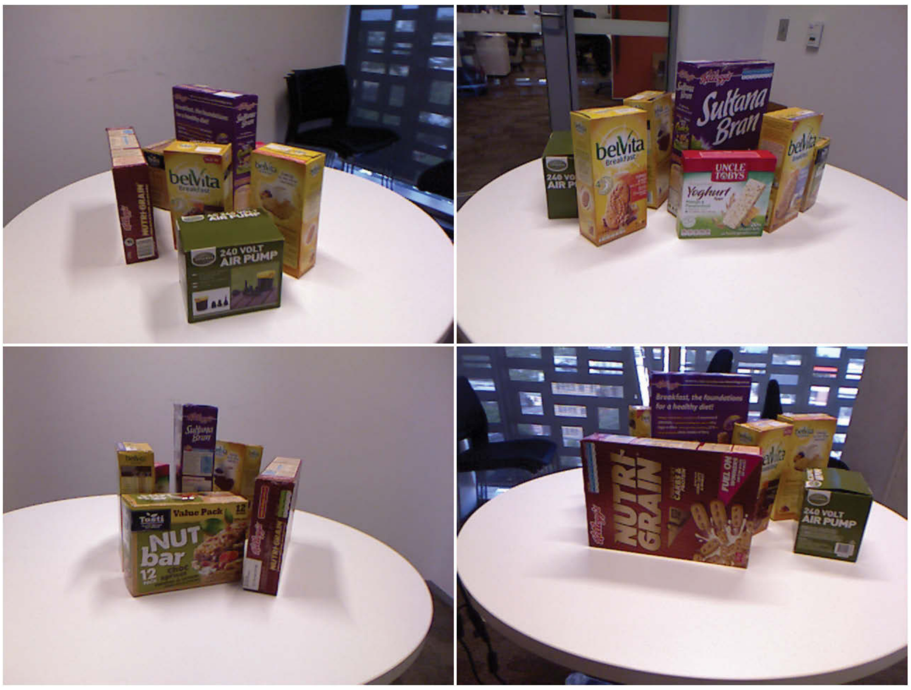


FIGURE 6.14: Example images from the pre-collected RGB-D data.

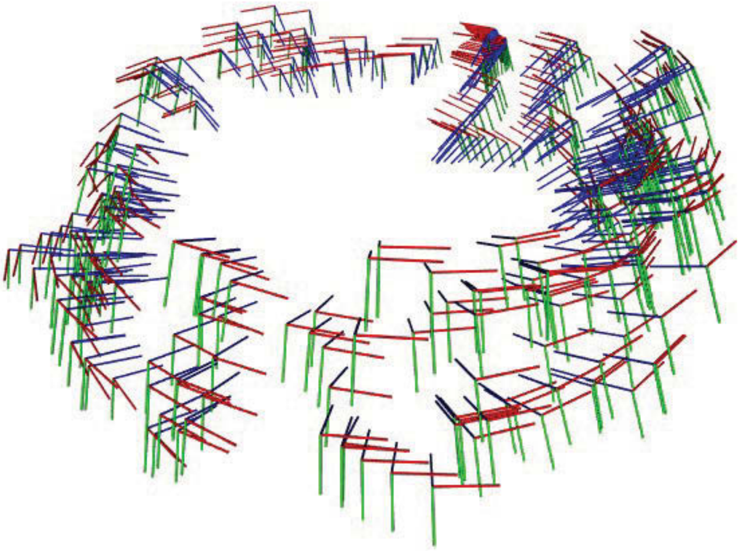
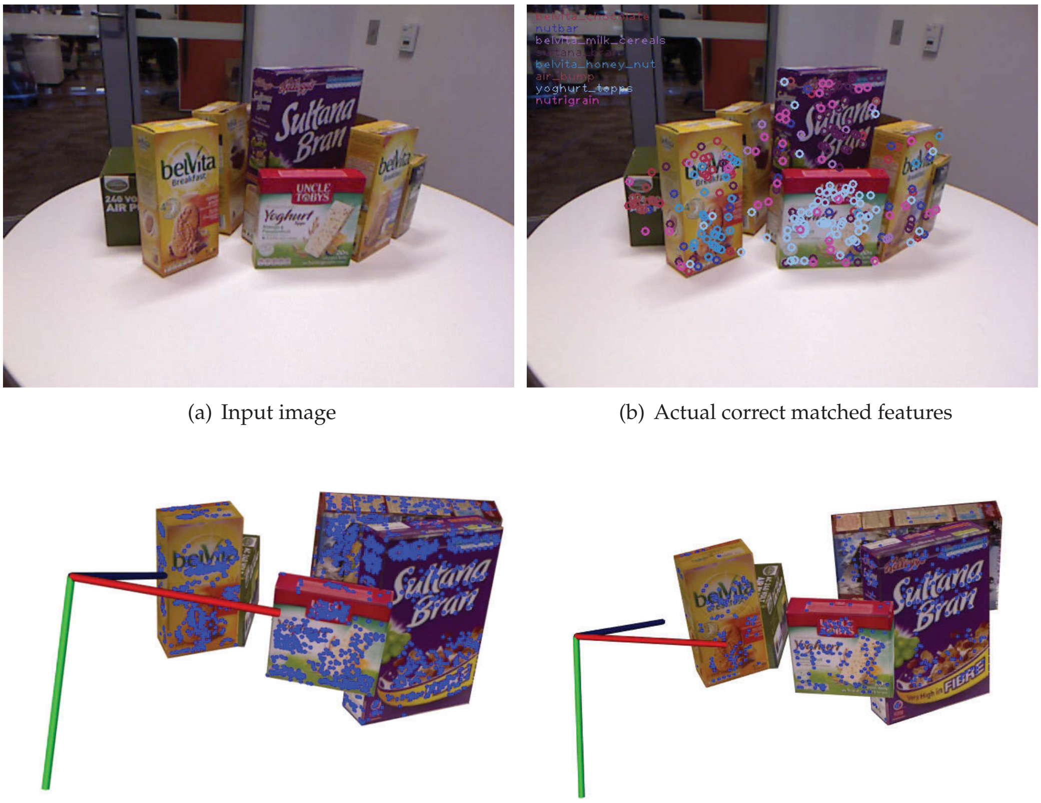


FIGURE 6.15: Optimised camera poses using RGBD-SLAM

Results and Discussion

1. **Prediction of Possible Matches:** Fig.6.16 shows the differences between prediction results with and without using $d_{\max}$ and $\alpha_{\max}$. Fig.6.16(a) is the actual acquired image in frame 2 and Fig.6.16(b) shows the matched features when the robot moves to the 2nd frame. If $d_{\max}$ and $\alpha_{\max}$, are not used, the prediction is not accurate enough as shown in Fig. 6.16(c). Similar to the conclusion in Section. 6.3.1, it can be seen that the predicted matches using the proposed strategy that uses these two additional information produces significantly improved prediction results as shown in Fig. 6.16(d).

2. **Planned Path and Reconstructed Scenario:** Using the collected data at every pose shown in Fig. 6.15, a path is generated which can cover the whole space on the table. The planned path is shown in Fig. 6.17. When the robot moves along this path, the objects can be recognised one-by-one and finally all of the objects on the table can be detected in spite of similar objects exist in the environment. Fig. 6.18 shows the reconstructed scene during different steps in the active object recognition. In the bottom-right figure, the ellipses highlight the detected *similar* objects using the newly introduced weight to each feature.



(a) Input image

(b) Actual correct matched features

(c) Predicted matches without using $d_{\max}$ and $\alpha_{\max}$

(d) Predicted matches using $d_{\max}$ and $\alpha_{\max}$

FIGURE 6.16: Object recognition and pose estimation results in different steps of the path

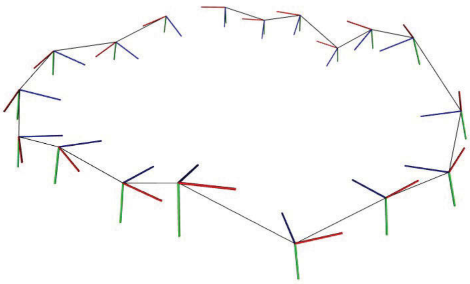


FIGURE 6.17: Planned trajectory for active object recognition and pose estimation

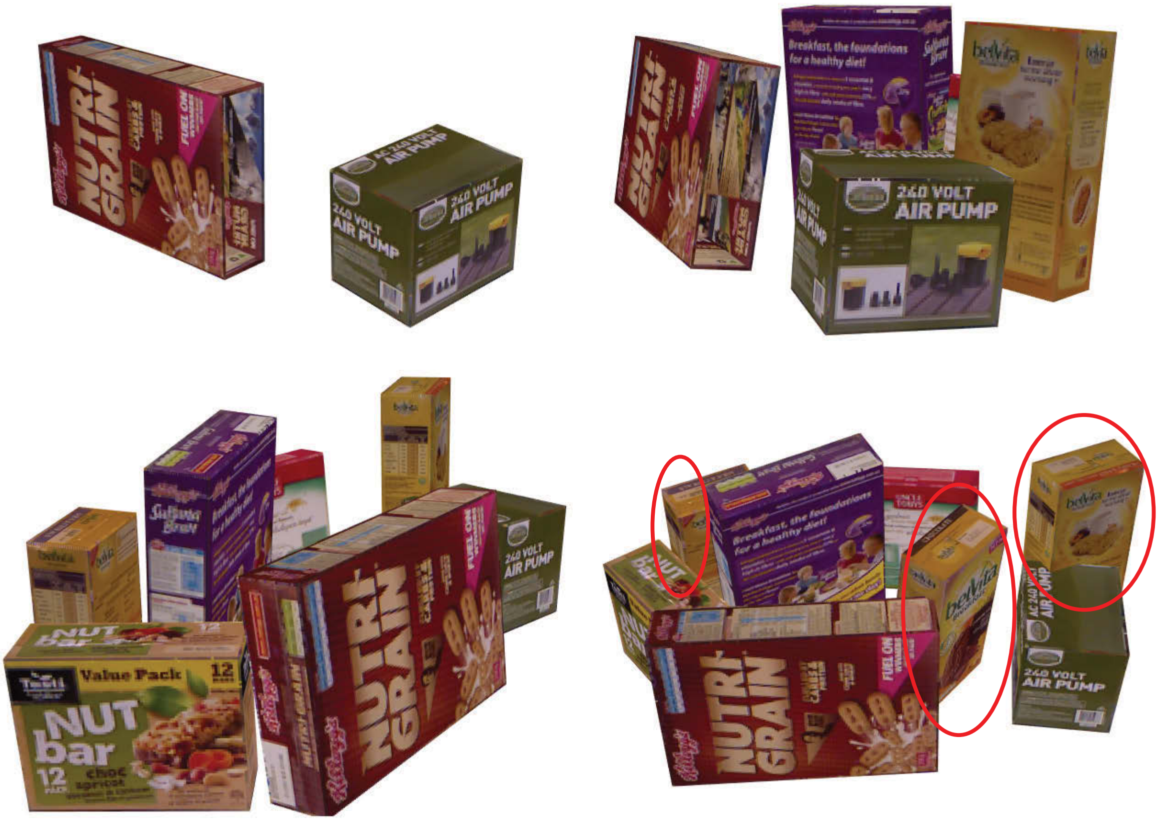


FIGURE 6.18: Object recognition and pose estimation results in different steps of the path.

Computational Cost

Approximate timing information for each of the steps in the algorithm during the robot trajectory in Fig. 6.17 is shown in Table. 6.1. The overall time consumption of recognition and estimation for each step is less than 1.5s even without using GPU computation. The voxelisation and the octree construction time only depend on the size of the work space. Therefore it is approximately the same during each iteration. Predicting matches for one voxel using raycasting, scale analysis and angle analysis takes less than 10 ms for about 3000 feature points. The overall prediction time depends on the number of voxels that are searched in the neighbourhood .

TABLE 6.1: Time consumption analysis for individual steps

Step name	Time (ms)[1]
RGB-D segmentation	745
Feature extracton	287
Feature matching	54
Consistent Matches searching	< 1
Pose estimation	< 1
Post-processing	< 1
Voxelisation and octree construction	~ 117
Overall	~ 1206

Unlike online active recognition in which all the working space is free to move, in these experiments, the trajectory can be generated given the positions of limited waypoints. In each step, instead of searching for the nearest free voxels, the method only searches the nearest neighbour points with pre-collected data.

6.4 Conclusion

In this chapter, an active object recognition and pose estimation system is presented which is able to localise cluttered household objects in the environment. By adding two more attributes, maximum observable distance and maximum observable angle, to the model, the proposed method is able to provide much more realistic prediction in Next-Best-View decision making. To discriminate similar objects, a feature weighting scheme is further proposed by using the distance from query feature to the closest different-class feature (called *near-miss*). The active recognition trajectory is generated by joining the nearest neighbour NBV problem into a consecutive process. A new object recognition and pose estimation system is also presented in this chapter. Via actively

[1] Intel(R) Core(TM) i7-3630QM CPU@2.40GHz

moving in the environment, the proposed system is able to cover all objects in the environment with accurate relative poses even with occlusion and ambiguities presented. With additional feature weights, the algorithm can generate trajectory which differentiates similar objects better.

In future work, the author is interested in taking motion uncertainty during viewpoint control into active object detection problem. This will help in implementing the active recognition system onto a less accurate robotic platform or without additional sensor positioning. Extending this one-step greedy NBV search algorithm into multiple steps is also another issue for future focus.

Chapter 7

Active Object Detection and Pose Estimation in Belief Space

In Chapter. 6, a model driven solution towards the active object detection and pose estimation problem is presented. Enriching the information in pre-trained object models enables more accurate predictions of the observable information under a new candidate viewpoint, thus laying the root for a successful active object detection framework even with a naive greedy search planning algorithm. However, in the above work, the motion model and observation model need to be perfect which is an unrealistic assumption for practical problems.

To address this limitation, this chapter presents a novel active object detection and pose estimation framework in belief space. The framework is able to incorporate various requirements for active object detection problem such as: object recognition confidence, pose estimate uncertainties for both robot and objects, and control consumptions. The solutions towards several critical issues during active object detection and pose estimation are presented, such as: 1) object pose initialisation; 2) initial guess for control optimisation; 3) collision avoidance; 4) online fast re-planning and 5) hypothesis changing during planning.

Throughout this chapter, bold lower-case and upper-case letters are reserved for vectors and matrices, respectively. Euclidean norm is denoted by $\|\cdot\|$. The weighted Euclidean norm of vector $\mathbf{e}$ with a positive definite matrix $\mathbf{W}$ is denoted by $\|\mathbf{e}\|_{\mathbf{W}} := \mathbf{e}^{\intercal}\mathbf{W}^{-1}\mathbf{e}$.

7.1 Preliminaries and Problem Formulation

7.1.1 Notation and Preliminaries

In this work, $\mathbf{p}_i^{\mathrm{o}}$ and $\mathbf{p}_k^{\mathrm{r}}$ are used to denote i-th object pose and robot pose at step k respectively. The state vector $\mathbf{X}_k$ consists of all previous robot poses $\mathbf{p}_{1:k}$ till step k and the poses of all detected objects, thus can be represented as

$$\mathbf{X}_k = \left[\mathbf{p}_1^{\mathrm{o}}, ..., \mathbf{p}_{n_o}^{\mathrm{o}}, \mathbf{p}_1^{\mathrm{r}}, ..., \mathbf{p}_k^{\mathrm{r}}\right] \tag{7.1}$$

Under the 2D cases, $\mathbf{p}^{\mathrm{o}} = [x^{\mathrm{o}}, y^{\mathrm{o}}, \theta^{\mathrm{o}}]$ and $\mathbf{p}_k^{\mathrm{r}} = [x_k^{\mathrm{r}}, y_k^{\mathrm{r}}, \theta_k^{\mathrm{r}}]$, n_o is the number of detected objects in state vector $\mathbf{X}_k \in \mathbb{R}^{3\times(n_o+k)}$.

$\mathbf{Z}_k$ is used to denote the observation at time step k. Under the assumption in [31, 160] that the object model consists of features on the object, the observation vector $\mathbf{Z}_k$ is a concatenation of the observed features on the detected objects. Assuming only *one* object is detected, observation $\mathbf{Z}_k$ is represented as $[\mathbf{z}_{k,1}, ..., \mathbf{z}_{k,n_k}]$ where n_k is the number of observed features on object at step k. In this work, a typical range-bearing sensor in 2D environment is adopted; therefore, $\mathbf{z}_{k,i}$ is explicitly represented $\left[x_{k,i}^z, y_{k,i}^z\right]$ which denotes the coordinate of feature f_i in current sensor frame by assuming that the sensor frame and robot frame coincide.

Given the notations of state vector $\mathbf{X}_k$ and observation $\mathbf{Z}_k$ above, the probabilistic motion model and observation model are formulated as

$$\begin{aligned} p(\mathbf{X}_{k+1}|\mathbf{X}_k, u_k) \Rightarrow & \mathbf{X}_{k+1} = \mathbf{F}(\mathbf{X}_k, u_k) + \eta_k \\ p(\mathbf{Z}_k|\mathbf{X}_k) \Rightarrow & \mathbf{Z}_k = \mathbf{H}(\mathbf{X}_k) + \xi_k \end{aligned} \tag{7.2}$$

By adapting the Markov assumption which has been widely accepted in SLAM community[40] and also by assuming that the robot is moving in a static environment, the motion and observation models with additive Gaussian noise are re-written as

$$\begin{aligned} \mathbf{p}_{k+1}^{\mathrm{r}} =& f(\mathbf{p}_k^{\mathrm{r}}, u_k) + \eta_k, \ \eta_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_\eta\right) \\ \mathbf{z}_{k,i} =& h(\mathbf{p}_k^{\mathrm{r}}, \mathbf{p}^{\mathrm{o}}, f_i^{\mathrm{o}}) + \xi_k, \ \xi_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_\xi\right) \end{aligned} \tag{7.3}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{I}\right)$ denotes a Gaussian random variable $\boldsymbol{\epsilon}$ with mean $\boldsymbol{\mu}$ and information matrix $\mathbf{I}$. f_i^o is the coordinate of feature i in object frame.

Given observations up to k-th step, $\mathbf{Z}_{1:k}$, and control input up to $(k-1)$-th step, $u_{1:k-1}$, the probability distribution function of state vector is represented as

$$p\left(\mathbf{X}_k|\mathbf{Z}_{1:k}, u_{1:k-1}\right) \tag{7.4}$$

This is a similar formulation from the traditional SLAM problem[39] as well. However, compared with feature based SLAM, a significant difference is that instead of inserting every feature(landmark) estimate into the state vector, a more compact representation is provided using the object pose $\mathbf{p}^o$ and the coordinates of features in object frame which are provided as object model information. More importantly, as a planning problem, the future observations are not given, and the core problem is to find the optimal control from a specifically designed objective function which will be explained later in this section.

7.1.2 Problem statement

The objective of this work is to present a planning strategy for the active object detection and pose estimation problem which allows a robot to autonomously explore the environments, recognise the target objects and estimate their relative poses. Model Predictive Control(MPC) framework is adopted into the system. At time step k, an optimal control strategy, $u^{\triangle}_{k:k+L-1} = \left\{u^{\triangle}_{k}, ..., u^{\triangle}_{u+L-1}\right\}$, in L steps horizon is computed via optimising the objective function $J_k\left(u_{k:k+L-1}\right)$ at time step k and the effectiveness of the generated trajectory is evaluated by the objective function J.

7.1.3 Formulation

Within planning horizon $[1, L]$, the generalised belief $\mathbf{b}\left(X_{k+l}\right)$ at l-th step is defined as

$$\mathbf{b}\left(\mathbf{X}_{k+l}\right) \doteq p\left(\mathbf{X}_{k+l}|\mathbf{Z}_{1:k}, u_{1:k-1}, \mathbf{Z}_{k+1:k+l}, u_{k:k+l-1}\right) \tag{7.5}$$

In (7.5), the observations up to l-th step $\mathbf{Z}_{1:k+l}$ and control inputs $u_{1:k+l}$ are partitioned into 2 parts: 1) control input $u_{1:k-1}$ and observation $\mathbf{Z}_{1:k}$ until step k which are available and 2) $u_{k:k+l-1}$ and $\mathbf{Z}_{k+1:k+l}$ which can only be predicted in l planning steps ahead. The general belief $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$ is an extension of standard belief space $\mathbf{b}\left(\mathbf{X}_{k}\right)$ into future l-th steps. $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$ is also assumed to follow a Gaussian distribution as below

$$\mathbf{b}\left(\mathbf{X}_{k+l}\right) \sim \mathcal{N}\left(\mathbf{X}_{k+l}^{\Delta}, \mathbf{I}_{k+l}\right) \tag{7.6}$$

where $\mathbf{X}_{k+l}^{\Delta}$ coincides with the *Maximum A Posteriori* (MAP) estimate of the $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$

$$\begin{aligned} \mathbf{X}_{k+l}^{\Delta} &= \arg\max_{\mathbf{X}_{k+l}} \mathbf{b}\left(\mathbf{X}_{k+l}\right) \\ &= \arg\min_{\mathbf{X}_{k+l}} -\log \mathbf{b}\left(\mathbf{X}_{k+l}\right) \end{aligned} \tag{7.7}$$

and $\mathbf{I}_{k+l}$ is the corresponding information matrix. Even though the mixture of Gaussians may be a better model of the belief[48], however, the uni-modal Gaussian distribution is still a widely-accepted and realistic assumption, such as [124] and [116]. The major difficulty in estimating the mean and information matrix of $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$ lies in the unknown future control $u_{k:k+l-1}$ and observation $\mathbf{Z}_{k+1:k+l}$ which will be addressed in Section. 7.2.

Once the belief $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$ is given, the objective function can be constructed and here a generalised formulation of the objective function is presented

$$J_k\left(u_{k:k+L-1}\right) \doteq \mathbb{E}\left\{\sum_{l=0}^{L-1} c_l\left(\mathbf{b}\left(\mathbf{X}_{k+l}\right), u_{k+l}\right) + c_L\left(\mathbf{b}(\mathbf{X}_{k+L})\right)\right\} \tag{7.8}$$

where $c_l\left(\cdot\right)$ is the intermediate cost function which counts both belief $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$ and control input u_{k+l} when l is in $[1, L-1]$ and c_L is the final state cost function which is only parameterised on $\mathbf{b}\left(\mathbf{X}_{k+L}\right)$. The optimal control input $u_{k:k+L-1}^{\Delta}$ is computed as (7.9) which is also the objective of this paper.

$$\begin{aligned} u_{k:k+L-1}^{\Delta} &\doteq \left\{u_k^{\Delta}, ..., u_{k+L-1}^{\Delta}\right\} \\ &= \arg\min_{u_{k:k+L-1}} J_k\left(u_{k:k+L-1}\right) \end{aligned} \tag{7.9}$$

7.2 Viewpoints Planning in General Belief Space

In order to optimise the objective function shown in (7.8), given the initial value of control inputs $u_{l:k+L-1}^{0}$, the general belief $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$ needs to be inferred firstly. Secondly, the optimised control $u_{l:k+L-1}^{\Delta}$ needs to be computed via optimising the objective function which is parameterised on $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$. This section first illustrates the inference of the general belief and control optimisation scheme. The formulation of the objective function is then presented for the active object detection and pose estimation problem. As a critical component in the objective function, *feature association probability*, is also explained and modelled in this section.

7.2.1 MAP Estimation in General Belief Space

Given the formulation of the general belief in (7.5), $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$ can be decomposed up to current belief $\mathbf{X}_k$ as

$$\begin{aligned}\mathbf{b}\left(\mathbf{X}_{k+l}\right) \doteq & p\left(\mathbf{X}_{k+l}|\mathbf{Z}_{1:k}, u_{1:k-1}, \mathbf{Z}_{k+1:k+l}, u_{k:k+l-1}\right) \\ \propto & p(\mathbf{X}_k|\mathbf{Z}_{1:k}, u_{1:k-1}) \prod_{i=1}^{l} \left[p\left(\mathbf{X}_{k+i}|\mathbf{X}_{k+i-1}, u_{k+i-1}\right) p\left(\mathbf{Z}_{k+i}|\mathbf{X}_{k+i}\right)\right]\end{aligned} \tag{7.10}$$

where the feature observation and association are assumed to be perfect in future steps. However, in practical problems, this assumption is too optimistic and also unrealistic. To model the feature association probability, a new variable $\gamma_{i,j}$ is introduced for observation $\mathbf{z}_{i,j}$ which represents a correct association of feature f_j^o at step i. Therefore, the probabilistic observation model is extended from $p\left(\mathbf{Z}_{k+i}|\mathbf{X}_{k+i}\right)$ into

$$\begin{aligned}p\left(\mathbf{Z}_{k+i}, \Gamma_{k+i}|\mathbf{X}_{k+i}\right) &= \prod_{j=1}^{n_i} p\left(\mathbf{z}_{k+i,j}, \gamma_{k+i,j}|\mathbf{X}_{k+i}\right) \\ &= \prod_{j=1}^{n_i} \left(p\left(\mathbf{z}_{k+i,j}|\mathbf{X}_{k+i}, \gamma_{k+i,j}\right) p\left(\gamma_{k+i,j}|\mathbf{X}_{k+i}\right)\right)\end{aligned} \tag{7.11}$$

where $p\left(\gamma_{k+i,j}|\mathbf{X}_{k+i}\right)$ describes the *feature association probability* given state $\mathbf{X}_{k+i}$. If the feature can be observed and given the association probability, $p\left(\mathbf{z}_{k+i,j}|\mathbf{X}_{k+i}, \gamma_{k+i,j}\right)$ describes the probability of capturing observation $\mathbf{z}_{k+i,j}$ given state $\mathbf{X}_{k+i}$ and $\gamma_{k+i,j}$

Via introducing parameter γ, (7.5) is marginalised on the latent variables $\Gamma_{k+1:k+l}$ as follows

$$\mathbf{b}\left(\mathbf{X}_{k+l}\right) \doteq \sum_{\Gamma_{k+1:k+l}} p\left(\mathbf{X}_{k+l}, \Gamma_{k+1:k+l}|\mathbf{Z}_{1:k}, u_{1:k-1}, \mathbf{Z}_{k+1:k+l}, u_{k:k+l-1}\right) \tag{7.12}$$

and the MAP estimate of $\mathbf{b}\left(\mathbf{X}_{k+l}\right)$ is computed as

$$\begin{aligned}\mathbf{X}^{\Delta}_{k+l} =& \underset{\mathbf{X}_{k+l}}{\arg\min} \underset{\Gamma_{k+1:k+l}|\bar{\mathbf{X}}_{k+l}}{\mathbb{E}} \left[-\log p\left(\mathbf{X}_{k+l}, \Gamma_{k+1:k+l}|\mathbf{Z}_{1:k}, u_{1:k-1}, \mathbf{Z}_{k+1:k+l}, u_{k:k+l-1}\right)\right] \\ =& \underset{\mathbf{X}_{k+l}}{\arg\min} \underset{\Gamma_{k+1:k+l}|\bar{\mathbf{X}}_{k+l}}{\mathbb{E}} \left[-\log p(\mathbf{X}_k|\mathbf{Z}_{1:k}, u_{1:k-1}) \right.\\ & \left. \prod_{i=1}^{l}\left(p\left(\mathbf{X}_{k+i}|\mathbf{X}_{k+i-1}, u_{k+i-1}\right)\prod_{j=1}^{n_i}\left(p\left(\mathbf{z}_{k+i,j}|\mathbf{X}_{k+i}, \gamma_{k+i,j}\right)p\left(\gamma_{k+i,j}|\mathbf{X}_{k+i}\right)\right)\right)\right]\end{aligned} \tag{7.13}$$

Recalling the Gaussian observation and motion models, (7.13) is further re-written as

$$\begin{aligned}\mathbf{X}^{\Delta}_{k+l} =& \underset{\mathbf{X}_{k+l}}{\arg\min} ||\mathbf{X}_k - \mathbf{X}^{\Delta}_k||^2_{\mathbf{I}_k} + \sum_{i=1}^{l}||\mathbf{X}_{k+i} - \mathbf{F}\left(\mathbf{X}_{k+i-1}, u_{k+i-1}\right)||^2_{\mathbf{I}_\eta} + \\ & \sum_{i=1}^{l}\sum_{j=1}^{n_i} p\left(\gamma_{k+i,j}|\mathbf{X}_{k+i}\right)||\mathbf{z}_{k+i,j} - h\left(\mathbf{X}_{k+i}, f^{\mathrm{o}}_j\right)||^2_{\mathbf{I}_\xi} \\ =& \underset{\mathbf{X}_{k+l}}{\arg\min} ||\mathbf{X}_k - \mathbf{X}^{\Delta}_k||^2_{\mathbf{I}_k} + \sum_{i=1}^{l}||\mathbf{X}_{k+i} - \mathbf{F}\left(\mathbf{X}_{k+i-1}, u_{k+i-1}\right)||^2_{\mathbf{I}_\eta} + \\ & \sum_{i=1}^{l}\sum_{j=1}^{n_i}||\mathbf{z}_{k+i,j} - h\left(\mathbf{X}_{k+i}, f^{\mathrm{o}}_j\right)||^2_{\bar{\mathbf{I}}^{i,j}_\xi}\end{aligned} \tag{7.14}$$

where $\bar{\mathbf{I}}^{i,j}_\xi = p\left(\gamma_{k+i,j}|\mathbf{X}_{k+i}\right)\mathbf{I}^{i,j}$ and $\mathbf{X}^{\Delta}$ is the current estimate of $\mathbf{X}_k$.

In (7.14), the MAP estimate of $\mathbf{X}_{k+l}$ is computed by minimising three terms:

- $||\mathbf{X}_k - \mathbf{X}^{\Delta}_k||^2_{\mathbf{I}_k}$: the uncertainty of current state vector $\mathbf{X}_k$;
- $\sum_{i=1}^{l}||\mathbf{X}_{k+i} - \mathbf{F}\left(\mathbf{X}_{k+i-1}, u_{k+i-1}\right)||^2_{\mathbf{I}_\eta}$: the motion uncertainty given $u_{k:k+l-1}$ control inputs in the predicted steps;
- $\sum_{i=1}^{l}\sum_{j=1}^{n_i}||\mathbf{z}_{k+i,j} - h\left(\mathbf{X}_{k+i}, f^{\mathrm{o}}_j\right)||^2_{\bar{\mathbf{I}}^{i,j}_\xi}$: the observation uncertainty;

Up to this point, the original problem in (7.8) is re-written into (7.14) using Expectation-Maximisation[101] where the MAP estimate of the "future" state vector $\mathbf{X}_{k+l}$ can be computed via traditional Gaussian Newton optimisation scheme. (7.14) is first linearised on nominal state $\bar{\mathbf{X}}_{k+l}(u_{k:k+l-1})$

$$\begin{aligned}\mathbf{X}^{\Delta}_{k+l} =& \underset{\Delta\mathbf{X}_{k+l}}{\arg\min} \left\|\Delta\mathbf{X}_k\right\|^2_{\mathbf{I}_k} + \sum_{i=1}^{l}\left\|\Delta\mathbf{X}_{k+i} - \frac{\partial\mathbf{F}}{\partial\mathbf{X}_{k+i-1}}\Delta\mathbf{X}_{k+i-1}\right\|^2_{\mathbf{I}_\eta} + \\ & \sum_{i=1}^{l}\sum_{j=1}^{n_i}\left\|\frac{\partial h_j}{\partial\mathbf{X}_{k+i}}\Delta\mathbf{X}_{k+i} - b^h_{i,j}(\mathbf{z}_{k+i,j})\right\|^2_{\bar{\mathbf{I}}^{ij}_\xi}\end{aligned} \tag{7.15}$$

where $\Delta\mathbf{X}_k$ is the difference between two consecutive iterations and $b^h_{i,j}\left(\mathbf{z}_{k+i,j}\right) = h\left(\mathbf{X}_{k+i}, f^{\mathrm{o}}_j\right) - h\left(\bar{\mathbf{X}}_{k+i}, f^{\mathrm{o}}_j\right)$ and the linearisation point $\bar{\mathbf{X}}_{k+l}\left(u_{k:k+l-1}\right)$ is explicitly computed given control $u_{k:k+l-1}$ iteratively as (7.16)

$$\begin{aligned} &\bar{\mathbf{X}}_{k+l}\left(u_{k:k+l-1}\right) \\ =&\mathbf{F}\left(\bar{\mathbf{X}}_{k+l-1}, u_{k+l-1}\right) \\ =&\mathbf{F}\left(\mathbf{F}\left(\bar{\mathbf{X}}_{k+l-2}, u_{k+l-2}\right), u_{k+l-1}\right) \\ =&\mathbf{F}\left(\cdots\mathbf{F}\left(\mathbf{X}^{\Delta}_k, u_k\right)\cdots, u_{k+l-1}\right) \end{aligned} \tag{7.16}$$

The detail of the linearsation is illustrated in Appendix. A.1.

In order to compute the optimal $\Delta\mathbf{X}_{k+l}$, (7.15) is further converted into the following quadratic form and the detailed derivation is illustrated in Appendix. A.2

$$\left\|\mathcal{A}_{k+l}\Delta\mathbf{X}_{k+l} - \mathcal{B}_{k+l}\right\|^2 \tag{7.17}$$

where $\mathcal{A}_{k+l}$ is a function of $u_{k:k+l-1}$ and $\mathcal{B}_{k+l}$ is a function of both $u_{k:k+l-1}$ and future observations $\mathbf{Z}_{k+1:k+l}$. Based on normal equation, the update state vector $\Delta\mathbf{X}_{k+l}$ which minimises (7.16) is

$$\Delta\mathbf{X}_{k+l} = \left(\mathcal{A}^{\intercal}_{k+l}\mathcal{A}_{k+l}\right)^{-1}\mathcal{A}^{\intercal}_{k+l}\mathcal{B}_{k+l} \tag{7.18}$$

and nominal state vector is updated as

$$\mathbf{X}^{\Delta}_{k+l} \doteq \bar{\mathbf{X}}_{k+l} + \Delta\mathbf{X}_{k+l} \tag{7.19}$$

The information matrix is also updated as

$$\mathbf{I}_{k+l} \doteq \mathcal{A}^{\intercal}_{k+l}\mathcal{A}_{k+l} \tag{7.20}$$

So far, the Gaussian distribution is parameterised in general belief $\mathbf{b}(\mathbf{X}_{k+l})$ with the mean (7.19) and the information matrix (7.20).

7.2.2 Empirical Modelling of Feature Association

In (7.11), an additional term $p\left(\gamma_{k+i,j}|\mathbf{X}_{k+i}\right)$ is introduced which describes the possibility of a correct feature association of feature j at step $k+i$. In this subsection, based on previous empirical analysis shown in Fig. 6.3, $\bar{\gamma}_{k+i,j} = p\left(\gamma_{k+i,j} | \mathbf{X}_{k+i}\right)$ is approximated as the production of two Gaussian distributions considering two factors as below

- **Scale variations**: $\bar{\gamma}_{i,j}$ is assumed to show the maximum value when the feature is re-observed under distance μ_{γ_d} and follows a Gaussian distribution w.r.t the variation of the distance;

- **Viewpoint variations**: $\bar{\gamma}_{i,j}$ is also assumed to show the maximum value when the feature is re-observed at viewpoint angle μ_{γ_θ} and there is a Gaussian distribution when the viewpoint angle varies;

Note here that the Gaussian distribution may not be the most realistic assumption of the distribution. For example, in Jia et al.'s work[72], the heavy tailed *Gamma-compound-Laplace* is regarded as a better option. However, our formulation is not limited to Gaussian distribution and any other alternative distributions can be adopted. Just for easier annotation and computation, the Gaussian distribution is assumed in this work. Based on the above assumption, $\gamma_{i,j} = \hbar_j\left(X_i\right)$ is modelled explicitly as below:

$$\begin{aligned}\bar{\gamma}_{k+i,j} =& p\left(\gamma_{k+i,j} | \mathbf{X}_{k+i}\right)\\ =& \left(\frac{1}{\sigma_{\gamma_d}\sqrt{2\pi}}\exp\left(-\frac{\left(d_f-\mu_{\gamma_d}\right)^2}{2\sigma_{\gamma_d}}\right)\right)\left(\frac{1}{\sigma_{\gamma_\theta}\sqrt{2\pi}}\exp\left(-\frac{\left(\theta_f-\mu_{\gamma_\theta}\right)^2}{2\sigma_{\gamma_\theta}}\right)\right)\end{aligned} \tag{7.21}$$

where μ_{γ_d} and μ_{γ_θ} denote the means as was illustrated before, and σ_{γ_d} and σ_{γ_θ} denote the standard deviations respectively.

Take SIFT feature as an example, μ_{γ_d} represents the distance where the feature shows the maximum response and μ_{γ_θ} is the direction of the surface normal vector of the feature (assuming the descriptor is captured perpendicular to the surface of the feature). Larger $\sigma_{\gamma_\mathrm{d}}$ and σ_{γ_θ} allow the feature to be correctly matched under larger range. d_f denotes the current distance between $(x_i^\mathrm{r}, y_i^\mathrm{r})$ and the feature. θ_f denotes the angle between the ray from the feature to $(x_i^\mathrm{r}, y_i^\mathrm{r})$ and μ_{γ_θ}. Please notice that both d_f and θ_f are calculated from $\mathbf{X}_i$ and the pre-trained information shows the coordinate and normal vector of feature in object coordinate frame.

Fig. 7.1 shows an example of the feature correspondence score distribution where the red arrow denotes the normal vector and the green line draws the surface. The red peak in Fig. 7.1 represents the area where the feature has a higher possibility of being correctly matched.

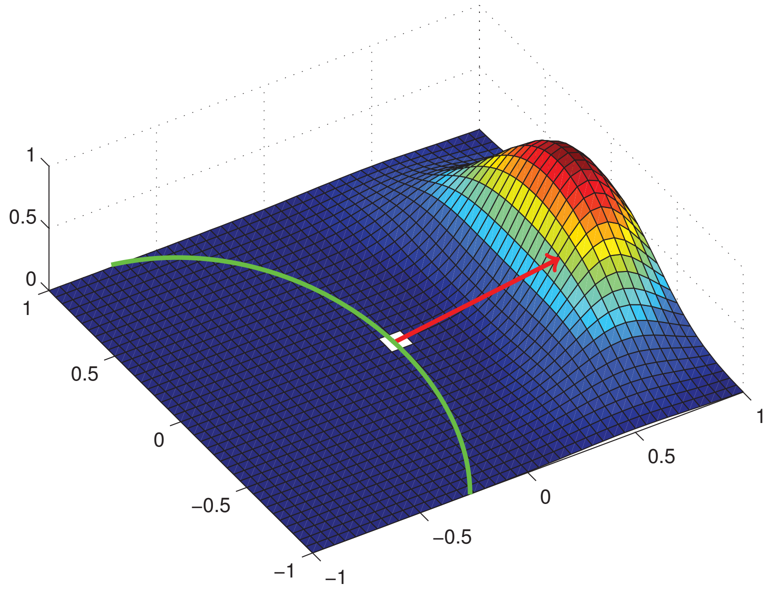


FIGURE 7.1: Probabilistic distribution of feature correspondence confidence

7.2.3 Optimal Control Inference

In practical problems, the Field-of-View of the sensor will lead to the discontinuity of the objective function $J_k\left(u_{k:k+L-1}\right)$ together with occlusion scenarios which will be discussed later in Section.7.3.3. In this case, in order to compute the optimal control $u^{\triangle}_{k:k+L-1}$ which satisfies

$$u^{\triangle}_{k:k+L-1} = \underset{u_{k:k+L-1}}{\arg\min} J_k(u_{k:k+L-1}) \tag{7.22}$$

A Derivative Free Optimisation (DFO) method, Nelder Mead Simplex-Reflective method[108], is adopted rather than using gradient based methods where derivatives are approximated using central difference. The implementation and simulation in Section. 7.4 are using `fminsearchbnd` in Matlab[1].

Given initial guess $u^{(0)}_{k:k+L-1}$, the algorithm first creates the elements of the simplex around $u^{(0)}_{k:k+L-1}$ by adding 5% on each component of $u^{(0)}_{k:k+L-1}$ to the original value. Using the created elements and $u^{(0)}_{k:k+L-1}$, a simplex is formed and the algorithm modifies and shrinks the simplex iteratively until it converges. By denoting $\mathbf{x}\left(i\right), i = 1, ..., n+1$ as the points on the simplex, the

[1] The original `fminsearch` function from Matlab does not support bounded constraint on the parameters $u_{k:k+L-1}$ and a modified `fminsearchbnd` from D'Errico is used. This modification is achieved by rewriting the parameter using $\sin\left(\right)$ function which is bounded within $[-1, 1]$

detailed algorithm is illustrated in Algorithm. 3 as below

Algorithm 3: Nelder-Mead Simplex algorithm for control optimisation

```
while not converged do
    Order $\mathbf{x}(i)$ from lowest function value $J_k(\mathbf{x}(1))$ to highest $J_k(\mathbf{x}(n+1))$;
    Generate reflected point as;
        $\mathbf{r} = 2\bar{\mathbf{x}} - \mathbf{x}(n+1)$;
    where $\bar{\mathbf{x}} = \frac{\sum_{i=1}^{n} \mathbf{x}(i)}{n}$ and calculate $J_k(\mathbf{r})$;
    if $J_k(\mathbf{x}(1)) \leq J_k(\mathbf{r}) \leq J_k(\mathbf{x}(k))$ then
        $\mathbf{x}(n+1) = \mathbf{r}$;
    else if $J_k(\mathbf{r}) < J_k(\mathbf{x}(1))$ then
        $\mathbf{s} = \bar{\mathbf{x}} + 2(\bar{\mathbf{x}} - \mathbf{x}(n+1))$;
        if $J_k(\mathbf{s}) < J_k(\mathbf{r})$ then
            $\mathbf{x}(n+1) = \mathbf{s}$;
        else
            $\mathbf{x}(n+1) = \mathbf{r}$;
        end
    else if $J_k(\mathbf{r}) > J_k(\mathbf{x}(n))$ then
        if $J_k(\mathbf{r}) < J_k(n+1)$ then
            $\mathbf{c} = \bar{\mathbf{x}} + (\mathbf{r} - \bar{\mathbf{x}})$;
            if $J_k(\mathbf{c}) > J_k(\mathbf{r})$ then
                $\mathbf{x}(n+1) = \mathbf{c}$
            else
                $\mathbf{v}(i) = \mathbf{x}(1) + \frac{(\mathbf{x}(i) - \mathbf{x}(1))}{2}$ for $i = 2, .., n+1$;
                $\mathbf{x}(i) = \mathbf{v}(i)$ for $i = 2, ..., n+1$;
            end
        else
            $\mathbf{t} = \bar{\mathbf{x}} + \frac{\mathbf{x}(n+1) - \bar{\mathbf{x}}}{2}$;
            if $J_k(\mathbf{t}) < J_k(\mathbf{x}(n+1))$ then
                $\mathbf{x}(n+1) = \mathbf{t}$;
            else
                $\mathbf{v}(i) = \mathbf{x}(1) + \frac{(\mathbf{x}(i) - \mathbf{x}(1))}{2}$ for $i = 2, .., n+1$;
                $\mathbf{x}(i) = \mathbf{v}(i)$ for $i = 2, ..., n+1$;
            end
        end
end
```

7.2.4 Objective Function Formulation

In order to design a comprehensive objective function for active object detection and pose estimation tasks, the following issues at least have to be considered with the declining priority

1. **maximise** the object recognition confidence by observing the object from different viewpoints;

2. **minimise** the uncertainty of the pose estimation of *both* the robot and the target objects while the robot explores the environment;

3. **minimise** the control consumption to achieve a smooth trajectory;

Apparently, in practical problems, more factors are required to be included such as occlusion and obstacle avoidance. However, these issues will be discussed in the forthcoming Section. 7.3. Regarding the 3 issues above, the objective function is formulated as

$$\begin{aligned} J_k\left(u_{k:k+L-1}\right) \doteq & w_f^J \left\{\mathbf{c}_f\left(\mathbf{X}_k^{\Delta}\right) - \mathbb{E}[\mathbf{c}_f\left(\mathbf{X}_{k+L}^{\Delta}\right)]\right\} + \\ & w_X^J \mathbf{c}_X\left(\mathbf{b}\left(\mathbf{X}_{k+L}^{\Delta}\right)\right) + \\ & w_u^J \mathbf{c}_u\left(u_{l:k+L-1}\right) \end{aligned} \tag{7.23}$$

The 3 terms in the above (7.23) are explained in detail below sequentially

1. The evaluation of the object recognition confidence is a non-trivial problem here. In computer vision communities where researchers adopt machine learning techniques heavily[121, 56], criteria such as precision, recall and mean Average Precision are utilised to describe the capability of a classifier in differentiating the objects. However, since the traditional point-feature based object recognition framework is used, these criteria are not suitable for our problem, therefore, the following assumption is proposed

   > The object recognition confidence is increased if 1) *larger number of features* are observed or 2) one feature is observed in *larger number of frames*.

   Intelligibly speaking, the above assumption means that, to confirm an object, observing 2 features provides more information about the object compared with observing only 1 feature and observing 1 feature for 2 times is better than observing the same feature for only once. This is a realistic and also understandable assumption. Given this assumption, the *observation score matrix* $\mathbf{O}(\mathbf{X}^{\Delta}_{k+L})$ at step $k+L$ is defined as

$$\mathbf{O}(\mathbf{X}^{\Delta}_{k+L}) = \left.\underbrace{\begin{bmatrix} {}^1o_{1,1} & \cdots & {}^1o_{1,k} & \cdots & {}^1o_{1,k+L} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^1o_{n_1,1} & \cdots & {}^1o_{n_1,k} & \cdots & {}^1o_{n_1,k+L} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^jo_{1,1} & \cdots & {}^jo_{1,k} & \cdots & {}^jo_{1,k+L} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^jo_{n_j,1} & \cdots & {}^jo_{n_j,k} & \cdots & {}^jo_{n_j,k+L} \end{bmatrix}}_{\text{number of steps: } k+L}\right\} \sum_{i=1}^{j} n_i \tag{7.24}$$

where ${}^io_{j,k}$ denotes the *feature association probability* of feature j of object i at step k and ${}^io_{j,k}$ is a function of $\mathbf{p}^{\mathrm{r}}_k$ and $\mathbf{p}^{\mathrm{o}}_i$[2]. ${}^io_{j,k}$ denotes the possibility of the feature being correctly associated at step k. Now the matrix $\mathbf{O}\left(\mathbf{X}^{\Delta}_{k+L}\right)$ which describes the feature association probability till step $k+L$ is computed and it needs to be summarised into a scalar value which can be minimised through the objective function. Hence the *object observation score* is formulated as a function of $\mathbf{O}\left(\mathbf{X}^{\Delta}_{k+L}\right)$ below

$$\begin{aligned} \rho =& \mathbf{c}_f\left(\mathbf{X}^{\Delta}_{k+L}\right) \\ =& \sum_{i=1}^{n_o}\sum_{j=1}^{n_i}\left(1 - \prod_{m=1}^{k+L}\left(1 -^i o_{j,m}\right)\right) \end{aligned} \tag{7.25}$$

(7.25) provides a solution to describe the object recognition confidence by considering the coverage of the features on the objects and the repeatability of the feature being observed.

To explain this formulation clearer, given an example of a 3×3 matrix $\mathbf{O}\left(\mathbf{X}_3\right) = \begin{bmatrix} 0.8 & 0.9 & 0.7 \\ 0.9 & 0.85 & 0.7 \\ 0.8 & 0.8 & 0.9 \end{bmatrix}$ (3 steps with 3 observed features), the possibility of feature 1 being missing after 3 observations is computed as a product of $(1-0.8)\times(1-0.9)\times(1-0.7) = 0.006$. Therefore, after 3 observations, the possibility of confirming this feature is $1 - 0.006 = 0.994$. By summing up all 3 features, ρ is equal to $0.994 + 0.9955 + 0.996 = 2.9855$.

By computing the difference between current *object observation score* calculated from $\mathbf{X}^{\Delta}_k$ and

[2] Strictly speaking, ${}^io_{j,k}$ should be written as ${}^io_{j,k}\left(\mathbf{X}^{\Delta}_k\right)$, however, here a simplified notation is used.

the one computed from the MAP estimate, $\mathbf{X}_{k+L}^{\triangle}$ from (7.19), it is possible to describe the gained information about the features on the objects from step k to step $k+L$. Therefore, minimising $\mathbf{c}_f\left(\mathbf{X}_k^{\triangle}\right) - \mathbb{E}[\mathbf{c}_f\left(\mathbf{X}_{k+L}^{\triangle}\right)]$ will maximise the number of covered features and the repeatability of each observed feature.

However, there are two points which needs to be highlighted about this term:

- Here, the *object observation score* is formulated as (7.25), however, the matrix formulation of (7.24) allows flexible extensions and modifications of score ρ. For example, based on the importance of each feature, it is possible to assign different weight to the score ${}^{i}o_{j,k}$; ρ can be also computed as

$$\rho = \sum_{i=1}^{n_o}\sum_{j=1}^{n_i}\left(\max_{m=1,.,k+L}\ {}^{i}o_{j,m}\right) \tag{7.26}$$

  where the maximum probability score is selected to represent the feature association probability after *a series* of observations;

- Please be aware that this term is actually computed as the difference $\mathbf{c}_f\left(\mathbf{X}_k^{\triangle}\right) - \mathbb{E}[\mathbf{c}_f\left(\mathbf{X}_{k+L}^{\triangle}\right)]$ where all history feature association probability is taken into consideration. This characteristic makes this objective function no longer following the Markov assumption where the previous information actually interferes with the future planning. From another perspective, this is an essential and realistic formulation where previous observation knowledge of the objects does *need* to influence the future planning trajectory.

2. $\mathbf{c}_X\left(\mathbf{b}\left(\mathbf{X}_{k+L}\right)\right)$ denotes the uncertainties of both the robots and the objects. As formulated in (7.1), $\mathbf{X}_{k+L}$ consists of object poses and all previous poses of the robot. In order to describe the uncertainty, this term is explicitly modelled as

$$\mathbf{c}_X\left(\mathbf{b}\left(\mathbf{X}_{k+L}\right)\right) \doteq \text{Tr}\left(\mathbf{W}_X \mathbf{I}_{k+L}^{-1} \mathbf{W}_X^{\intercal}\right) \tag{7.27}$$

where $\text{Tr}\left(\mathbf{M}\right)$ computes the *trace* of the matrix $\mathbf{M}$ and $\mathbf{W}_X$ is the weight matrix. In the simulation experiments in Section. 7.4, the uncertainty of the target objects and imminent steps are paid more attention to thus larger weights are assigned to close steps and smaller weights are assigned to further away steps.

3. $\mathbf{c}_u\left(u_{l:k+L-1}\right)$ represents the control consumption. In 2D environments such as the simulations in Section. 7.4, the control input u_{k+i} consists of velocity μ_{k+i} and angular velocity

ω_{k+i}. As was illustrated in Section. 7.2.3, both μ_{k+i} and ω_{k+i} are bounded within a specific range. In our simulations, this term is represented as

$$\mathbf{c}_u\left(u_{l:k+L-1}\right) \doteq \begin{cases} \mathbf{w}_u \boldsymbol{\omega}_{k:k+L-1} & \text{if } \mu_i \text{ are all positive or negative} \\ \mathbf{w}_u \boldsymbol{\omega}_{k:k+L-1} + \mathbf{c}_\mu & \text{else} \end{cases} \tag{7.28}$$

where $\boldsymbol{\omega}_{k:k+L-1} = [\omega_k, \omega_{k+1}, ..., \omega_{k+L-1}]^\intercal$ and $\mathbf{w}_u$ assigns different weight to different steps which plays the same role as $\mathbf{W}_X$ in (7.27). The principle of (7.28) is that if the robot plans a trajectory where all linear velocities are either positive or negative, the control cost is defined as the weighted sum of the angular velocity. However, if the robot plans a trajectory where linear velocity can be both negative and positive, a fixed, extra term $\mathbf{c}_\mu$ will be given to penalise the zigzag movements of the robot.

Among the detailed formulation of each term in the objective function (7.23), the main difficulty is to compute the last term $\mathbb{E}[\mathbf{c}_f\left(\mathbf{X}^\Delta_{k+L}\right)]$ where $\mathbf{X}^\Delta_{k+L}$ is related with the unknown future observations as shown in (7.18) and (7.19). To predict the future observations, traditional approaches[116][48] assume the future observations will be the same as their maximum likelihood estimates. However, there are two pieces of work[70, 149] which solve this problem via formulating the objective function ingeniously. In Van Den Berg and et al.'s work[149], the value function $v_t[\mathbf{b}]$ is approximated in a quadratic form below(Please refer to Section. 4.2 in [149] for detailed derivation.):

$$v_t[\mathbf{b}] \approx \frac{1}{2}\left(\mathbf{b} - \bar{\mathbf{b}}_t\right)^\intercal \mathbf{S}_t \left(\mathbf{b} - \bar{\mathbf{b}}_t\right) + \left(\mathbf{b} - \bar{\mathbf{b}}_t\right)\mathbf{s}_t + s_t \tag{7.29}$$

where $\bar{\mathbf{b}}_t$ is the nominal belief. In Indelman et al.'s work[70], the term which is related with the mean of future belief is formulated in the linear relationship as below(Please see Section. 5 in [70] for detailed information):

$$\mathbf{c}_L\left(\mathbf{X}_{k+L}\right) = \left\|\mathbf{E}^G_{k+L}\mathbf{X}^\Delta_{k+L} - \mathbf{X}^G\right\|^2_{\mathbf{M}_X} \tag{7.30}$$

where $\mathbf{E}^G_{k+L}$ is a selection matrix and $\mathbf{X}^G$ is the goal state. In the above two formulations, the formula below is used:

$$\mathop{\mathbb{E}}_{\mathbf{y}}[\mathbf{y}^\intercal\mathbf{Q}\mathbf{y}] = \mathop{\mathbb{E}}_{\mathbf{y}}[\mathbf{y}]^\intercal \mathbf{Q} \mathop{\mathbb{E}}_{\mathbf{y}}[\mathbf{y}] - \mathrm{Tr}\left(\mathbf{Q}\Sigma_{\mathbf{y}}\right) \tag{7.31}$$

where $\mathbf{y}$ is a random vector with covariance matrix $\Sigma_{\mathbf{y}}$. $\mathbf{Q}$ is a given matrix. Now assume the $\mathbf{y}$ denotes the uncertain noise in the future observations, (7.31) allows eliminating the mean of $\mathbf{y}$

and only keeping the covariance term which is given in the observation model.

Back to our problem, the last term $\mathbf{c}_f\left(\mathbf{X}^{\Delta}_{k+L}\right)$ is highly nonlinear. There are two ways to solve this problem

1. following the *maximum likelihood observation* assumption;

2. approximating the nonlinear term to the first or second order;

If the *maximum likelihood observation* is assumed, β in (A.9) is set as $\mathbf{0}$. An intuition to understand this assumption is that the predicted future observation will be exactly the same with captured actual observation when estimating the posterior belief. In the experiments which will be presented Section. 7.4, due to the complexity in approximating this nonlinear term into second-order, this work **follows** the *maximum likelihood observation* assumption.

7.2.5 Online Re-planning

In the current approach which is formulated in an MPC framework, the control optimisation step discussed in Section. 7.2.3 and Section. 7.2.4 is the most time-consuming step. Therefore, there is a need to re-plan the trajectory efficiently by utilising the optimisation results from previous steps. In this subsection, a fast re-planning strategy is presented by comparing the prior and posterior belief using KL-divergence.

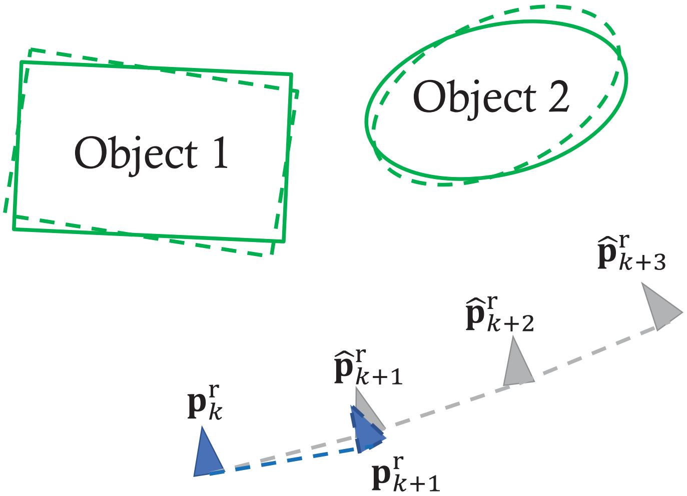

Object 1
Object 2
$\hat{\mathbf{p}}^{\mathrm{r}}_{k+3}$
$\hat{\mathbf{p}}^{\mathrm{r}}_{k+2}$
$\hat{\mathbf{p}}^{\mathrm{r}}_{k+1}$
$\mathbf{p}^{\mathrm{r}}_{k}$
$\mathbf{p}^{\mathrm{r}}_{k+1}$

FIGURE 7.2: Example: path planning for next 3 steps and the updated pose after 1 step of control execution.

Following the strategy of MPC, the following $L-1$ controls $u_{k:k+L-1}$ is computed from step k to $k+L$ given the current belief of the state. For example, assuming L is equal to 3, the optimised control $\mathbf{u}^{\triangle}_{k:k+2}$ will lead the robot to poses $\hat{\mathbf{p}}^{\mathrm{r}}_{k+1:k+3}$, shown as the gray triangles in Fig. 7.2. Afterwards, the first optimised control $\mathbf{u}^{\triangle}_{k}$ is executed and after acquiring the updated observations from the two objects, the estimate of the robot is computed as $\mathbf{p}^{\mathrm{r}}_{k+1}$, shown as the blue triangles in Fig. 7.2. The prior estimate of the objects (1 and 2) are denoted by the dashed shapes in Fig. 7.2 and the posterior estimate of the objects are shown as the shapes with solid border lines. Due to the noises in motion model and observation model, there will be a difference between the prior belief and posterior belief.

In order to fully utilise the previous optimisation results $u^{\triangle}_{k+1:k+2}$, a re-planning strategy using the Kullback-Leibler divergence is presented to compare the difference between two distributions as below:

$$\mathbf{b}_{k+1} \sim \mathcal{N}\left(\mathbf{X}_{k+1}, \mathbf{I}_{k+1}\right)\ , \hat{\mathbf{b}}_{k+1} \sim \mathcal{N}\left(\hat{\mathbf{X}}_{k+1}, \hat{\mathbf{I}}_{k+1}\right) \tag{7.32}$$

where $\hat{\mathbf{b}}_{k+1}$ denotes the prior belief of object poses $\mathbf{p}^{\mathrm{o}}_{1}, \mathbf{p}^{\mathrm{o}}_{2}$ and $\mathbf{p}^{\mathrm{r}}_{k+1}$ obtained from the prediction at step $k+1$ and $\mathbf{b}_{k+1}$ is the posterior belief of the same variables after executing the control and capturing the observation. The KL divergence is computed as below:

$$\begin{aligned} D_{\mathrm{KL}}(\mathbf{b}_{k+1} \| \hat{\mathbf{b}}_{k+1}) =& \frac{1}{2} \log \frac{|\hat{\mathbf{I}}^{-1}_{k+1}|}{|\mathbf{I}^{-1}_{k+1}|} - \frac{1}{2} \mathrm{Tr}\left\{\mathbf{I}_{3(n_o+k+1)}\right\} + \\ & \frac{1}{2}\left(\mathbf{X}_{k+1} - \hat{\mathbf{X}}_{k+1}\right)^{\intercal} \hat{\mathbf{I}}_{k+1} \left(\mathbf{X}_{k+1} - \hat{\mathbf{X}}_{k+1}\right) + \frac{1}{2}\mathrm{Tr}\left\{\hat{\mathbf{I}}_{k+1} \cdot \mathbf{I}^{-1}_{k+1}\right\} \end{aligned} \tag{7.33}$$

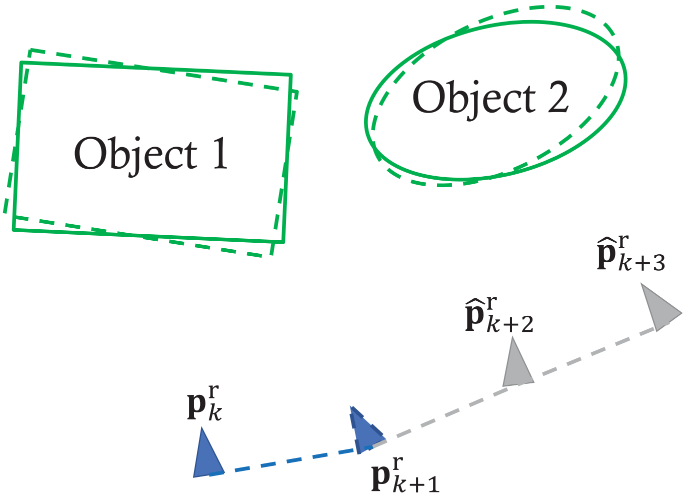

Object 1
Object 2
$\hat{\mathbf{p}}^{\mathrm{r}}_{k+3}$
$\hat{\mathbf{p}}^{\mathrm{r}}_{k+2}$
$\mathbf{p}^{\mathrm{r}}_{k}$
$\mathbf{p}^{\mathrm{r}}_{k+1}$

FIGURE 7.3: Replanning strategy if the belief is only changed within the threshold.

KL divergence describes the similarity between two distributions. Less $D_{\mathrm{KL}}(\mathbf{b}_{k+1} \| \hat{\mathbf{b}}_{k+1})$ denotes larger similarity between two beliefs $\mathbf{b}_{k+1}$ and $\hat{\mathbf{b}}_{k+1}$ and vice versa. As shown in Fig. 7.3, if $D_{\mathrm{KL}}(\mathbf{b}_{k+1} \| \hat{\mathbf{b}}_{k+1})$ is less than a given threshold ρ_{KL}, the next pose is set as $\hat{\mathbf{p}}^{\mathrm{r}}_{k+1}$ which is computed from the previous optimised controls thus avoiding the redundant optimisation steps.

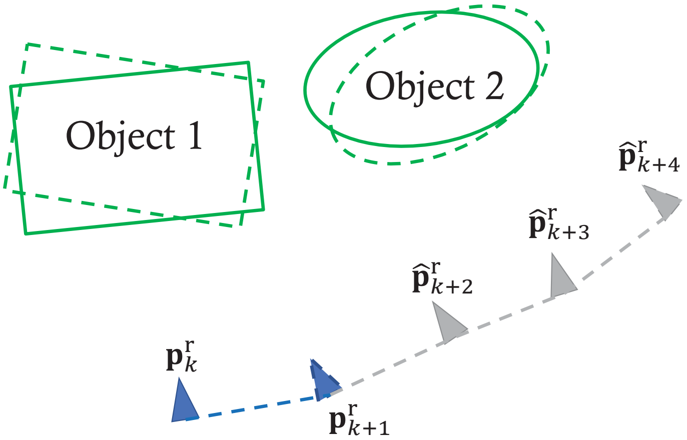

Object 1
Object 2
$\hat{\mathbf{p}}^{\mathrm{r}}_{k+4}$
$\hat{\mathbf{p}}^{\mathrm{r}}_{k+3}$
$\hat{\mathbf{p}}^{\mathrm{r}}_{k+2}$
$\mathbf{p}^{\mathrm{r}}_{k}$
$\mathbf{p}^{\mathrm{r}}_{k+1}$

FIGURE 7.4: Replanning strategy if the belief shows significant differences compared with the prediction.

If $D_{\mathrm{KL}}(\mathbf{b}_{k+1} \| \hat{\mathbf{b}}_{k+1}) > \rho_{\mathrm{KL}}$ where estimated poses of the objects and robot changes significantly as shown in Fig. 7.4, the previous belief $\hat{b}_{k+1}$ cannot be trusted after capturing the observation and updating the state. Therefore, it is necessary to re-do the initialisation step presented in Section. 7.3.2 and re-run the optimisation step thus abandoning the previous optimisation results. Moreover, the robot is forced to start sampling and optimisation once there is less than 2 optimised poses remaining in the experiments. During the experiments, this re-planning strategy significantly reduces the number of executions of the time-consuming optimisation step,

7.3 Practical Issues

7.3.1 Object Pose Initialisation

Under the proposed framework, when a target object is discovered, its relative pose and covariance w.r.t the world coordinate frame need to be estimated. This problem can appear at the beginning step or when a new object is discovered and added into the state vector during the planning phase. In order to deal with this problem, here the beginning step is used as an example to show how to compute the object's pose and covariance.

In this subsection, ${}_{\mathrm{r}}\mathbf{T}_{\mathrm{o}}$ is denoted as the object's pose w.r.t the camera frame, ${}_{\mathrm{w}}\mathbf{T}_{\mathrm{o}}$ as the object's pose w.r.t the world frame and ${}_{\mathrm{w}}\mathbf{T}_{\mathrm{r}}$ as the robot's pose w.r.t the world frame. Please also note that $\mathbf{p}^{\mathrm{o}}$ is equivalent to the estimate of ${}_{\mathrm{w}}\mathbf{T}_{\mathrm{o}}$ and $\mathbf{p}^{\mathrm{r}}$ in the state vector is the estimate of ${}_{\mathrm{w}}\mathbf{T}_{\mathrm{r}}$.

Assuming the observation model in (7.2), the object's pose w.r.t the object can be estimated using a Least Square Estimator. ${}_{\mathrm{r}}\mathbf{T}_{\mathrm{o}}$ is trivial to compute and the computed least square estimate is computed as ${}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}}$, the covariance matrix of the least square estimate is computed as

$$ {}_{\mathrm{r}}\Sigma_{\mathrm{o}} = \mathbf{H}^{\intercal}_{({}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}})}\boldsymbol{\Sigma}_{\eta}\mathbf{H}_{({}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}})} \tag{7.34} $$

where $\boldsymbol{\Sigma}_{\eta}$ is the block diagonal covariance matrix stacked by Σ_{η} from (7.2) and

$$ \mathbf{H}_{({}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}})} = \left.\frac{\partial h\left({}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}} + \Delta\delta\right)}{\partial\Delta\delta}\right|_{\Delta\delta=0} \tag{7.35} $$

Under 2D environments, $\mathbf{H}_{({}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}})} \in \mathbb{R}^{2n\times 3}$ and $\boldsymbol{\Sigma}_{\eta} \in \mathbb{R}^{2n\times 2n}$ where n is the number of observed features. So far, the pose of the object w.r.t the camera , ${}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}}$, and its covariance ${}_{\mathrm{r}}\Sigma_{\mathrm{o}}$ are provided.

Given initial robot pose ${}_{\mathrm{w}}\mathbf{T}^{\star}_{\mathrm{r}} = \mathbf{p}^{\mathrm{r}}_{1}$ and its covariance ${}_{\mathrm{w}}\Sigma_{\mathrm{o}}$, , now the state vector and the covariance need to be augmented as

$$ \begin{aligned} \bar{\mathbf{X}}_1 = [\mathbf{p}^{\mathrm{r}}_1] \in \mathbb{R}^3 &\Longrightarrow \mathbf{X}_1 = [\mathbf{p}^{\mathrm{o}}_1, \mathbf{p}^{\mathrm{r}}_1] \in \mathbb{R}^6 \\ \bar{\boldsymbol{\Sigma}}_1 = {}_{\mathrm{w}}\Sigma_{\mathrm{o}} \in \mathbb{R}^{3\times 3} &\Longrightarrow \boldsymbol{\Sigma}_1 = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \in \mathbb{R}^{6\times 6} \end{aligned} \tag{7.36} $$

In order to achieve the above augment, ${}_{\mathrm{w}}\mathbf{T}^{\star}_{\mathrm{r}}$ and ${}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}}$ in error state are computed as

$$ \begin{aligned} {}_{\mathrm{w}}\mathbf{T}_{\mathrm{r}} &= {}_{\mathrm{w}}\mathbf{T}^{\star}_{\mathrm{r}} \oplus {}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}} \\ &= {}_{\mathrm{w}}\mathbf{T}^{\star}_{\mathrm{r}} \cdot \mathrm{Vec2T}({}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}) \\ {}_{\mathrm{r}}\mathbf{T}_{\mathrm{o}} &= {}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}} \oplus {}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}} \\ &= {}_{\mathrm{r}}\mathbf{T}^{\star}_{\mathrm{o}} \cdot \mathrm{Vec2T}\left({}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}\right) \end{aligned} \tag{7.37} $$

where ${}_{\mathrm{w}}\mathbf{T}_{\mathrm{r}}$ and ${}_{\mathrm{r}}\mathbf{T}_{\mathrm{o}}$ in $\mathbb{R}^{3\times 3}$ are the groundtruth pose and ${}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}, {}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}} \in \mathbb{R}^3$ are the errors with zero means and covariance matrices ${}_{\mathrm{w}}\Sigma_{\mathrm{r}}$ and ${}_{\mathrm{r}}\Sigma_{\mathrm{o}}$ respectively. Vec2T is the function which converts the pose representation $\mathbf{p} = [x, y, \theta]$ into a homogeneous transformation matrix representation $\mathbf{T}$

as below

$$\mathbf{T} = \mathrm{Vec2T}\left(\mathbf{p}\right) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & x \\ \sin(\theta) & \cos(\theta) & y \\ 0 & 0 & 1 \end{bmatrix} \tag{7.38}$$

Therefore, the error state representation of ${}_{\mathrm{w}}\mathbf{T}_{\mathrm{o}}$ is

$${}_{\mathrm{w}}\mathbf{T}_{\mathrm{o}} = \left({}_{\mathrm{w}}\mathbf{T}_{\mathrm{r}}^{\star} \oplus {}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}\right) \cdot \left({}_{\mathrm{r}}\mathbf{T}_{\mathrm{o}}^{\star} \oplus {}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}\right) \tag{7.39}$$

and the Jacobian matrix w.r.t the error $\mathbf{e} = [{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}^{\mathsf{T}}, {}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}^{\mathsf{T}}]^{\mathsf{T}}$ and the first-order expansion are

$$\begin{aligned} \mathbf{F} &= \left.\frac{\partial {}_{\mathrm{w}}\mathbf{T}_{\mathrm{o}} \ominus \left({}_{\mathrm{w}}\mathbf{T}_{\mathrm{r}}^{\star} \cdot {}_{\mathrm{r}}\mathbf{T}_{\mathrm{o}}^{\star}\right)}{\partial \mathbf{e}}\right|_{\mathbf{e}=\mathbf{0}} \\ {}_{\mathrm{w}}\mathbf{T}_{\mathrm{o}} &= \left({}_{\mathrm{w}}\mathbf{T}_{\mathrm{r}}^{\star} \cdot {}_{\mathrm{r}}\mathbf{T}_{\mathrm{o}}^{\star}\right) \oplus \left(\mathbf{F} \cdot \mathbf{e}\right) \end{aligned} \tag{7.40}$$

So far, recalling (7.36), the object pose w.r.t the world frame, $\mathbf{p}_1^{\mathrm{o}}$, is computed as

$$\mathbf{p}_1^{\mathrm{o}} = {}_{\mathrm{w}}\mathbf{T}_{\mathrm{r}}^{\star} \cdot {}_{\mathrm{r}}\mathbf{T}_{\mathrm{o}}^{\star} \tag{7.41}$$

Considering the 4 blocks in the covariance matrix, firstly, the $\mathbf{D}$ is equal to the original covariance of the robot pose ${}_{\mathrm{w}}\Sigma_{\mathrm{r}}$ and $\mathbf{A}$ is calculated as

$$\begin{aligned} \mathbf{A} &= \mathbf{F} \begin{bmatrix} {}_{\mathrm{w}}\Sigma_{\mathrm{r}} & \\ & {}_{\mathrm{r}}\Sigma_{\mathrm{o}} \end{bmatrix} \mathbf{F}^{\mathsf{T}} \\ &= \begin{bmatrix} \mathbf{F}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}} \\ \mathbf{F}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}} \end{bmatrix} \begin{bmatrix} {}_{\mathrm{w}}\Sigma_{\mathrm{r}} & \\ & {}_{\mathrm{r}}\Sigma_{\mathrm{o}} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}} \\ \mathbf{F}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}} \end{bmatrix}^{\mathsf{T}} \\ &= \mathbf{F}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}} ({}_{\mathrm{w}}\Sigma_{\mathrm{r}}) \mathbf{F}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}}^{\mathsf{T}} + \mathbf{F}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}} ({}_{\mathrm{r}}\Sigma_{\mathrm{o}}) \mathbf{F}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}}^{\mathsf{T}} \end{aligned} \tag{7.42}$$

where $\mathbf{F}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}}$ and $\mathbf{F}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}}$ are from $\mathbf{F}$. After all, $\boldsymbol{\Sigma}_1$ is represented as

$$\begin{aligned}
\boldsymbol{\Sigma}_1 &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{F}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}} & \mathbf{F}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} {}_{\mathrm{r}}\Sigma_{\mathrm{o}} & \\ & {}_{\mathrm{w}}\Sigma_{\mathrm{r}} \end{bmatrix} \begin{bmatrix} \mathbf{F}^{\mathsf{T}}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}} & \mathbf{0} \\ \mathbf{F}^{\mathsf{T}}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}} & \mathbf{I} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{F}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}}({}_{\mathrm{w}}\Sigma_{\mathrm{r}})\mathbf{F}^{\mathsf{T}}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}} + \mathbf{F}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}}({}_{\mathrm{r}}\Sigma_{\mathrm{o}})\mathbf{F}^{\mathsf{T}}_{{}_{\mathrm{r}}\mathbf{e}_{\mathrm{o}}} & \mathbf{F}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}}{}_{\mathrm{w}}\Sigma_{\mathrm{r}} \\ {}_{\mathrm{w}}\Sigma_{\mathrm{r}}\mathbf{F}^{\mathsf{T}}_{{}_{\mathrm{w}}\mathbf{e}_{\mathrm{r}}} & {}_{\mathrm{w}}\Sigma_{\mathrm{r}} \end{bmatrix}
\end{aligned} \tag{7.43}$$

Up to now, given the estimated object's pose w.r.t robot and its covariance computed using (7.34) and (7.35), the method is able to augment the original state vector and covariance in (7.36) using (7.40) to (7.43).

7.3.2 Initial Guess for Control Optimisation

In order to obtain the optimal control $u^{\triangle}_{k:k+L-1}$ for objective function in (7.23), an initial value for the controls $u^{(0)}_{k:k+L-1}$ is required. To compute a reasonable initial value, it is necessary to sample the control series for future steps and select the best one as the initial guess for the control optimisation. The detailed steps are listed as below:

1. In this problem set-up, the control constraints of linear velocity are $[\nu_{\min}, \nu_{\max}]$, and the constraints of angular velocity are $[\omega_{\min}, \omega_{\max}]$ where $\nu_{\min} = -\nu_{\max}, \omega_{\min} = -\omega_{\max}$. Along each sampled trajectory which consists of L steps of control, it is required that all the linear velocity to be either all positive $\nu_{\max}$ or all negative $\nu_{\min}$ which implies the zigzag movements are not allowed during the sampling phase and the robot can only explore in *one* direction in *one* trajectory. To sample angular velocity control input for each step, a random number in range $[\omega_{\min}, \omega_{\max}]$ is generated. After all, 2×3^L series of sampled controls are computed;

2. From the sampled controls, the robot poses are generated from these controls. Given the current understanding of the environment which includes the detected objects and possible unknown obstacles, if the generated poses are located in the occupied areas, this trajectory is rejected from the candidates. After this step, 2×3^L control series are filtered into n control series where $n \leq 2 \times 3^L$;

3. Evaluating all n control series using the objective function and selecting the best one as the initial guess are extremely time-consuming. For example, in our experiments in Section. 7.4 where $L = 5$, there still will be hundreds of trajectories to be evaluated. Therefore, in order to speed up the algorithm, only m out of n controls series are selected where m is set to be 20 or 40 in Section. 7.4;

4. Evaluating the objective function on m control series and selecting the best one as the initial guess which will be input to the subsequent optimisation step;

7.3.3 Occlusion Modelling

Given the dense models of the objects as shown in Section. 7.1, self-occlusion and occlusion caused by other pre-trained objects can be predicted using raytracing algorithm given current belief of the state and the sensor's pose. However, in practical problems, the environment may consist of unknown obstacles and thus generate occlusions which interfere with future observations. Fig. 7.5 demonstrates an example of unknown occlusions as the shaded blue area. In order to solve this issue, an effective yet reasonable probabilistic modelling of the occlusions is presented.

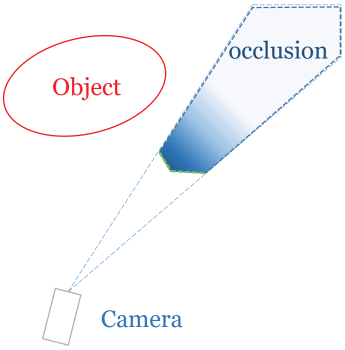
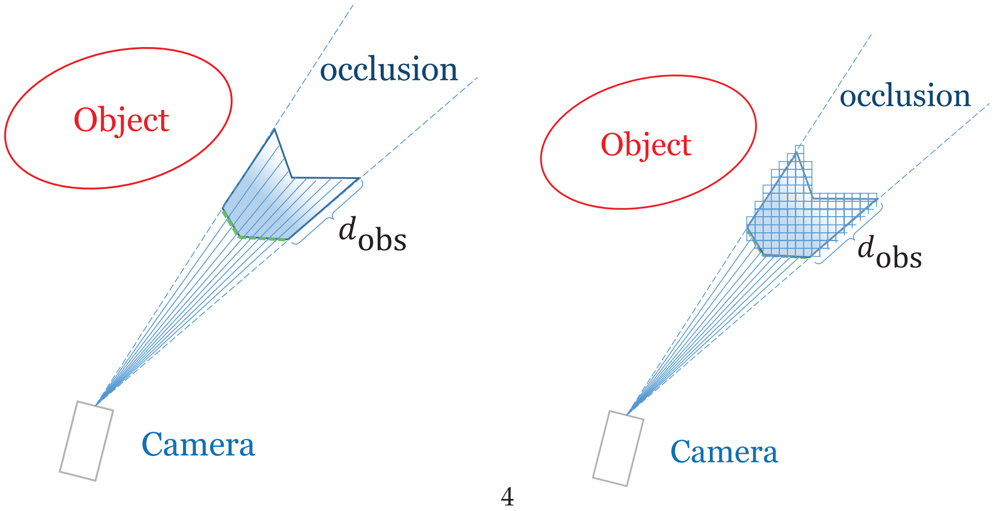


FIGURE 7.5: Occlusion modelling for observation prediction.

The fundamental assumption for modelling the occlusion is that the unobserved, occupied area behind the unknown obstacle has only a limited range of d_{obs} along the ray as shown in Fig. 7.6(a). The possibility of having occlusion is also assumed to be decreased along the ray in a linear relationship, shown as the shaded area in Fig. 7.6(b). Due to the fact that no prior knowledge of the unknown occlusion is given, this is a realistic assumption which means that the occlusion only happens within a limited hidden area behind the obstacle and areas which are further away behind the obstacle have less possibilities of being occupied.



(a) Occluded area modeling.

(b) Grid representation of the occlusions

FIGURE 7.6: Probabilistic occlusion modeling.

Following the above assumption, a grid map of the occluded area is generated as shown in Fig. 7.6(b) using the ray from the sensor to the points on the obstacle. A *occlusion possibility* ρ_{occ} is associated with each grid and further away grids are assigned with smaller ρ_{occ}. For example, d_{obs} is assumed to be $0.5m$ and there are 10 grids along the ray behind the obstacle. Given this assumption, the grid which is $0.25m$ far behind the obstacle will have $\rho_{\text{occ}} = 0.5$. In the prediction phase, as explained in Section. 7.2.4, this value can be incorporated seamlessly into *observation score matrix* $\mathbf{O}\left(\mathbf{X}^{\Delta}_{k+L}\right)$. If a feature is predicted to be occluded by a grid with *occlusion possibility* ρ_{occ}, each entry in $\mathbf{O}\left(\mathbf{X}^{\Delta}_{k+L}\right)$ is re-written as $\left(\rho_{\text{occ}} \cdot {}^{i}o_{j,k}\right)$. Taking an explicit case as an example to explain it more clearly, suppose the original value for ${}^{i}o_{j,k}$ is 0.8 which means that this feature has 80% probability of being associated correctly. If a grid with $\rho_{\text{occ}} = 0.8$ appears along the ray, new *feature observation score* is $0.8 * 0.8 = 0.64$ which means that due to the occlusion, this feature has only 64% probability to be observed and matched correctly.

7.3.4 Collision Avoidance

During the planning for active object detection and pose estimation, due to the complexity of the environment, another inevitable issue is to avoid the forbidden areas in the environment. Fig. 7.7 shows an example of the obstacles. The red ellipse shows a forbidden area such as the table where the object is placed on and the robot is not allowed to explore such a region. The right part

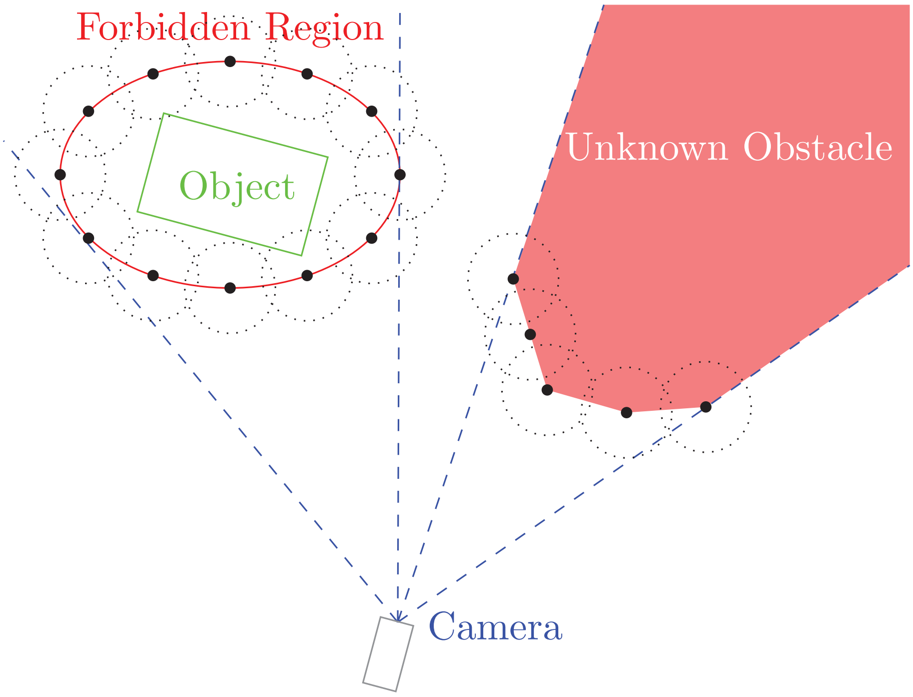


FIGURE 7.7: Obstacle avoidance in path planning.

shows an unknown obstacle and the shaded area can be the obstacle or not. Since the occlusion issue has been discussed in above Section. 7.3.3, here collision avoidance is the only focus.

Obstacle avoidance is another mature research topic in the robotic field. Lamiraux et. al [86] used an iterative scheme to find a collision free path using a potential field based on the obstacles. A more recently work, TrajOpt, from Schulman et. al[131] provides an elegant solution which penalises collisions with a hinge loss in a sequential convex optimisation framework. Here, an alternative strategy is provided that can be easily fitted into our framework. Points on the surface of the obstacles is sampled and the "forbidden" zone is defined as the circle area centred at each sampled point, shown as the black dashed circle in Fig. 7.7. If the robot falls into the "forbidden" zone, a larger penalty will be added into the objective function, thus having an additional term in the objective function as below

$$\mathbf{c}_{\text{obs}}\left(\mathbf{X}^{\Delta}_{k+L}\right) \doteq \sum_{k=1}^{L} \omega_{\text{obs}} \cdot \mathbf{1}\left(\mathbf{p}^{\text{r}}_{k+i}, \mathbf{Area}_{\text{obs}}\right) \tag{7.44}$$

where

$$\mathbf{1}\left(\mathbf{p}^{\text{r}}_{k+i}, \mathbf{Area}_{\text{obs}}\right) = \begin{cases} 1 & \mathbf{p}^{\text{r}}_{k+i} \text{ locates in occulusion regions } \mathbf{Area}_{\text{obs}} \\ 0 & \mathbf{p}^{\text{r}}_{k+i} \text{ locates in free regions} \end{cases} \tag{7.45}$$

and ω_{obs} is set to be a large value. Via minimising the objective function as below, it is able to avoid the obstacles in the environment.

$$J_k\left(u_{k:k+L-1}\right) \doteq \mathbf{c}_X\left(\mathbf{b}\left(\mathbf{X}^{\Delta}_{k+L}\right)\right) + \mathbf{c}_u\left(u_{l:k+L-1}\right) + \mathbb{E}[\mathbf{c}_f\left(\mathbf{X}^{\Delta}_{k+L}\right)] + \mathbb{E}\left[\mathbf{c}_{\text{obs}}\left(\mathbf{X}^{\Delta}_{k+L}\right)\right] \tag{7.46}$$

7.3.5 Hypothesis Changing during Planning

The previous Section. 7.2.5 discusses how to re-plan the trajectory when the poses and uncertainties are changed after acquiring actual observation while utilising the previous optimal control values. This section presents the solution when the object identity changes after capturing the latest observation. This also means that the previous object recognition hypothesis is incorrect and the identity of the object has to be changed as in the example below

$$\begin{aligned} \hat{\mathbf{X}}_{k+1} &= [\ \underset{\substack{\uparrow \\ \text{Object A}}}{\mathbf{p}^{\text{o}}_1}\ ,\ \underset{\substack{\uparrow \\ \text{Object B}}}{\mathbf{p}^{\text{o}}_2}\ , \mathbf{p}^{\text{r}}_1, ..., \mathbf{p}^{\text{r}}_k, \hat{\mathbf{p}}^{\text{r}}_{k+1}] \\ \Rightarrow \mathbf{X}_{k+1} &= [\ \underset{\substack{\uparrow \\ \text{Object A}}}{\mathbf{p}^{\text{o}}_1}\ ,\ {}^{\times}\underset{\substack{\uparrow \\ \text{Object C}}}{\mathbf{p}^{\text{o}}_2}, \mathbf{p}^{\text{r}}_1, ..., \mathbf{p}^{\text{r}}_k, \mathbf{p}^{\text{r}}_{k+1}] \end{aligned} \tag{7.47}$$

where the identity of $\mathbf{p}^{\text{o}}_2$ is switched from object B to object C. In order to complete the above conversion, this subsection refers to some fundamental knowledge in EKF-SLAM[40]. There are overall 4 steps to accomplish the conversion in (7.47) presented as below:

1. *state propagation*: In this step, using the available motion model and control input u^{Δ}_k, and using the formula below, it is able to compute the prior belief of the state as

$$\begin{aligned} \tilde{\mathbf{p}}^{\text{r}}_{k+1} = f\left(\mathbf{p}^{\text{r}}_k, u^{\Delta}_k, 0\right) &\Longrightarrow \tilde{\mathbf{X}}_K \doteq \left[\mathbf{p}^{\text{o}}_1, \mathbf{p}^{\text{o}}_2, \mathbf{p}^{\text{r}}_1, ..., \mathbf{p}^{\text{r}}_k, \tilde{\mathbf{p}}^{\text{r}}_{k+1}\right] \\ \tilde{\mathbf{\Sigma}}_{k+1} &= \mathbf{F}_{\mathbf{X}} \Sigma_k \mathbf{F}_{\mathbf{X}} + \mathbf{F}_\eta \mathbf{I}^{-1}_\eta \mathbf{F}^{\intercal}_\eta \end{aligned} \tag{7.48}$$

2. *state reduction*: After acquiring the observation at step $k+1$, from the feature association results, it is able to tell that the object hypothesis of object 2 should be item C rather than item B. Therefore, the corresponding entry of object 2 from the mean and covariance matrix needs to be removed.

Recalling the fundamental knowledge on statistics, given joint distribution of landmark $\mathbf{m}$ and state x

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_{\mathbf{m}} \end{bmatrix} \text{ and } \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{x\mathbf{m}} \\ \boldsymbol{\Sigma}_{\mathbf{m}x} & \boldsymbol{\Sigma}_{\mathbf{mm}} \end{bmatrix} \tag{7.49}$$

where $\boldsymbol{\mu}$ is the mean estimate and $\boldsymbol{\Sigma}$ is the covariance matrix. In (7.49), the mean of robot pose as $\boldsymbol{\mu}_x$ with covariance $\boldsymbol{\Sigma}_{xx}$ and mean of the landmark as $\boldsymbol{\mu}_{\mathbf{m}}$ with covariance $\boldsymbol{\Sigma}_{\mathbf{mm}}$ are computed. Therefore, the mean and covariance of object A and all robot poses from $\hat{\mathbf{X}}_k$ and $\hat{\mathbf{I}}_k$ are also extracted as:

$$\begin{aligned} \hat{\mathbf{X}}'_k &= [\mathbf{p}^{\mathrm{o}}_1, \mathbf{p}^{\mathrm{r}}_1, ..., \mathbf{p}^{\mathrm{r}}_k, \tilde{\mathbf{p}}^{\mathrm{r}}_{k+1}] \\ \hat{\boldsymbol{\Sigma}}'_{k+1} &= \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}^{\mathrm{oo}}_{1,1} & \tilde{\boldsymbol{\Sigma}}^{\mathrm{or}}_{1,1:k} \\ (\tilde{\boldsymbol{\Sigma}}^{\mathrm{or}}_{1,1:k})^{\intercal} & \tilde{\boldsymbol{\Sigma}}^{\mathrm{rr}}_{1:k,1:k} \end{bmatrix} \end{aligned} \tag{7.50}$$

where $\tilde{\boldsymbol{\Sigma}}^{\mathrm{oo}}_{1,1}$ is the covariance of object 1, $\tilde{\boldsymbol{\Sigma}}^{\mathrm{or}}_{1,1:k}$ is the block between object 1 and all poses $\mathbf{p}^{\mathrm{r}}_{1:k}$ and $\tilde{\boldsymbol{\Sigma}}^{\mathrm{rr}}_{1:k,1:k}$ is the covariance of all robot poses.

3. *state update and augment*: Given the observation of the object C, the mean estimate and covariance need to be augmented by estimating the pose of the object C. By conducting the object initialisation step explained in Section. 7.3.2. The state vector which includes new discovered is updated as

$$\mathbf{X}_k = [\mathbf{p}^{\mathrm{o}}_1, {}^{\times}\mathbf{p}^{\mathrm{o}}_2, \mathbf{p}^{\mathrm{r}}_1, ..., \mathbf{p}^{\mathrm{r}}_k, \mathbf{p}^{\mathrm{r}}_{k+1}] \tag{7.51}$$

where $\mathbf{p}^{\mathrm{o}}_1$ and $\mathbf{p}^{\mathrm{r}}_{1:k+1}$ is extracted from $\hat{\mathbf{X}}_{k+1}$ in (7.50). This is a similar task as object initialisation in Section. 7.3.1 where the only difference is that the object pose is computed w.r.t $\mathbf{p}^{\mathrm{r}}_{k+1}$ rather than $\mathbf{p}^{\mathrm{r}}_1$. The augment on covariance matrix also follows (7.43) in Section. 7.3.1.

7.4 Simulation Experiments and Results Analysis

This section presents simulation experiments and analysis to validate the effectiveness of the proposed framework. The structure of this section is summarised as below

1. Section. 7.4.1 illustrates the basic set-ups of the simulation experiments including the sensor model, motion, observation noise and control noise;

2. Section. 7.4.2 demonstrates the effectiveness and consistency of simplex optimisation method in this problem formulation even though it cannot guarantee a global optimal solution;

3. Section. 7.4.3 discusses the parameterisation in our problem including the planning horizon L and the weights in the objective function;

4. Section. 7.4.4 validates our approach under a more challenging scenario and the results are analysed;

5. Section. 7.4.5 validates the performance of our approach in environments which consist of both forbidden regions and unknown obstacles;

6. Section. 7.4.6 further demonstrates the effectiveness of the proposed approach when the object hypothesis is changed during the planning phase;

7.4.1 Experimental Set-up

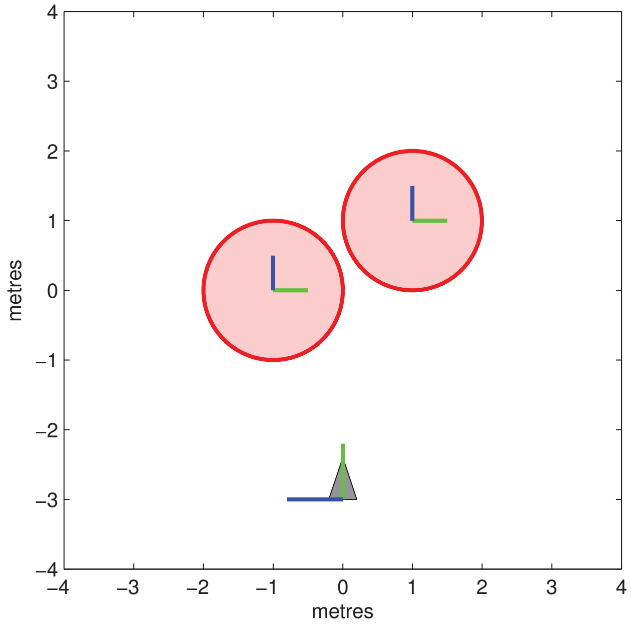


FIGURE 7.8: An example of the simulation environments.

Fig. 7.8 shows an example of the simulation environments for validating the proposed active object detection and pose estimation framework and this environment will be further tested in Section. 7.4.2 to Section. 7.4.3. The green and blue line segments denote the x axis and y

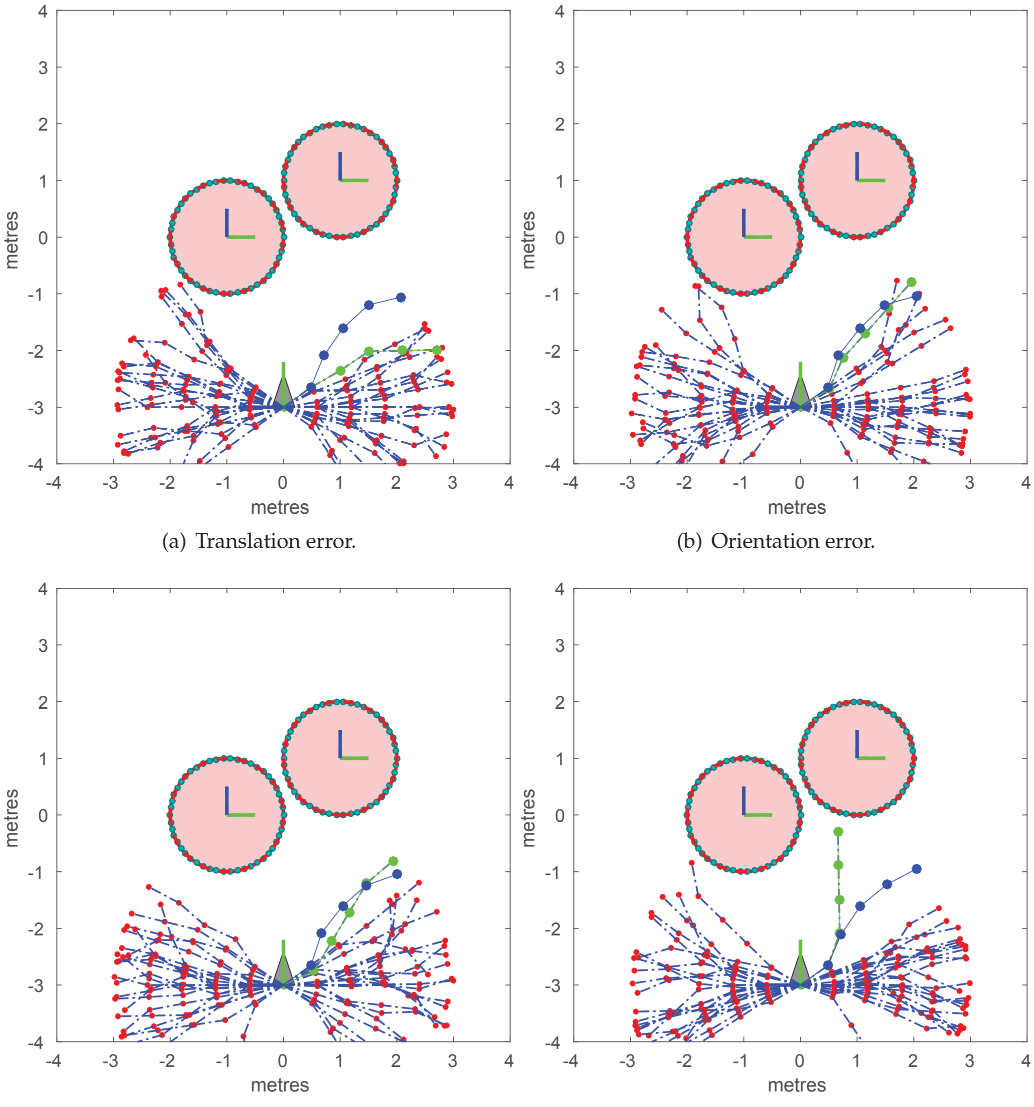
axis of the object frame and robot frame respectively. The robot, denoted as the gray triangle, is equipped with a bearing-range sensor with the Field-of-View equals to π and its range is between $0.5m \sim 4.0m$. The control inputs, linear velocity ν and angular velocity ω, are limited within $[-3, 3]$ m/s and $[-\pi, \pi]$ rad/s. The time duration of each step is 0.2 s which means that the maximum translation difference between two consecutive poses is less than $3 \times 0.2 = 0.6$ m and the maximum orientation difference is less than $\pi \times 0.2 = \frac{\pi}{5}$. The odometry noise is set as 5% of the input control values and the covariance of the observation noise is set as $\begin{bmatrix} 0.02^2 & 0 \\ 0 & 0.02^2 \end{bmatrix}$. Assume the size of the planning horizon L is set to be 5, the weights in planning phase, $\mathbf{W}_X$ and $\mathbf{w}_u$, are set as $\{1.0, 0.9, 0.8, 0.7, 0.6\}$ accordingly. Later Section. 7.4.3 will discuss further the parameterisation issues.

7.4.2 Convergence Analysis in Optimisation

Before presenting the planned trajectories using the proposed approach, in this section, since the simplex optimisation method is known as not being a global optimal solution, the convergence characteristics of the proposed approach is validated via extensive simulation experiments. Through comprehensive experiments, the effectiveness and convergence of the simplex method in this problem is demonstrated.

In this section, starting pose of the robot is $\left[0, -3, \frac{\pi}{2}\right]$. By assuming a perfect observation model without any noise, the optimised trajectories from different initial values under exactly the same belief are presented. Among all the figures which are presented in Fig. 7.9, the thin, blue and dash-dotted lines are the sampled trajectories, the green line is the best trajectory selected from the samples and set as the initial guess for the optimisation step, and the blue line is the trajectory generated from the optimised control series. There are two circular objects distributed in the environment with radius equals to 1m and the poses of the two objects are $[-1, 0, 0]$ and $[1, 1, 0]$. There are 50 features equally distributed on the surface of each object, shown as the green dots on the red circles which denote the objects.

The sampling strategiy as was illustrated in Section. 7.3.2 is adopted. Fig. 7.9(a) to Fig. 7.9(d) demonstrate the optimisation results under 40 randomly sampled trajectories. As the blue trajectories indicate, the optimisation results are different especially at future poses. As summarised in Table. 7.1, even though the values from the objective function are different. However, as highlighted in the red values, the first optimised poses are still very close to each other and the differences of linear velocity and angular velocity are limited with ±0.001m/s and ±0.005 rad/s. These errors in control input will lead to only 0.0002m translation error and $0.565°$ orientation error in the first step planned poses.



(a) Translation error.

(b) Orientation error.

(c) Translation error.

(d) Orientation error.

FIGURE 7.9: Optimisation results for different random sampled initial guess.

From the above experiments, it is observed that the *simplex* method cannot guarantee converging to exactly the same values, as indicated in Table. 7.1. However, by comparing the first planned pose and even the second planned pose, the differences between optimisation results from different initial values are constrained within 0.003m in translation and $4.586°$ in orientation. Therefore, simplex method is a suitable optimisation method in our framework.

TABLE 7.1: Optimisation results for randomly sampled initial guess

		Fig. 7.9(a)	Fig. 7.9(b)	Fig. 7.9(c)	Fig. 7.9(d)
Objective function value		-12.161	-8.799	-9.464	-7.135
Step 1	ν_1	-3.000	-2.998	-3.000	-2.998
	ω_1	3.141	3.132	3.130	3.141
Step 2	ν_2	-3.000	-2.998	-3.000	-2.987
	ω_2	2.897	3.099	3.135	2.671
Step 3	ν_3	-3.000	-2.971	-3.000	-3.000
	ω_3	-1.325	-1.716	-1.725	-0.903
Step 4	ν_4	-3.000	-2.998	-2.742	-2.977
	ω_4	-0.762	-0.664	-0.949	-1.427
Step 5	ν_5	-3.000	-2.998	-2.888	-2.981
	ω_5	-2.879	-2.511	-1.786	-0.979

7.4.3 Parameterisation

Planning Horizon L

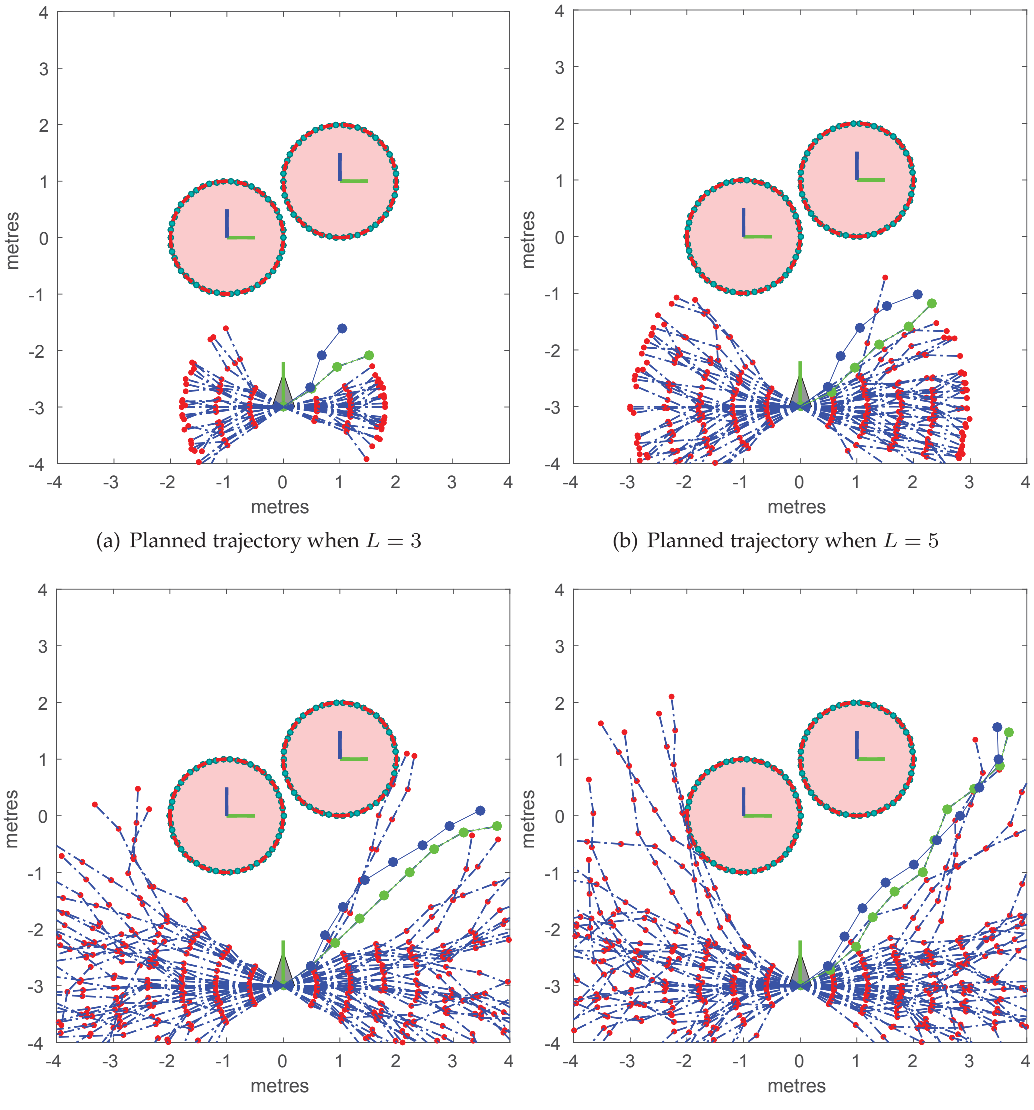
Among the parameters which need to be tuned, the size of planning horizon, L, significantly influences the timing consumption of the planning step. In short, larger L enables the robot to consider further away future steps by sacrificing the efficiency of the planning phase. In this section, the time consumption analysis is demonstrated with different numbers of $L = 3, 5, 8$ and 10 and the optimal value is selected based on comparative experiments and empirical analysis.

With larger planning horizon L, as shown in Fig. 7.10, the planned trajectories show better results and cover the object better. After 100 Monte Carlo simulations, the average time consumptions for 1 step of planning when $L = 3, 5, 8, 10$ are $8.209, 22.613, 43.858$ and 96.924 seconds respectively. By balancing the planning results and time consumption, in later experiments, L is set to be 5. Please note that $L = 5$ is the optimal solution from empirical analysis *only* under current simulation environments and other values may show better performances in different sized environments.

Weight Selection in Objective Function

In the objective function as below

$$\begin{aligned} J_k\left(u_{k:k+L-1}\right) \doteq & w_X^J \mathbf{c}_X\left(\mathbf{b}\left(\mathbf{X}^{\Delta}_{k+L}\right)\right) + \\ & w_u^J \mathbf{c}_u\left(u_{l:k+L-1}\right) + \\ & w_f^J \left\{\mathbf{c}_f\left(\mathbf{X}^{\Delta}_{k}\right) - \mathbb{E}[\mathbf{c}_f\left(\mathbf{X}^{\Delta}_{k+L}\right)]\right\} \end{aligned} \tag{7.52}$$



(a) Planned trajectory when $L = 3$

(b) Planned trajectory when $L = 5$

(c) Planned trajectory when $L = 8$

(d) Planned trajectory when $L = 10$

FIGURE 7.10: Comparative experiments of different L values.

without considering the obstacle penalty term, there are 3 weights which need to be tuned carefully during implementation. Here the comparison experiments of the combinations of parameters w_X^j, w_u^J and w_f^J are presented in Table. 7.2.

TABLE 7.2: Different weight parameterisation methods in the objective function.

Parameters	w_X^J	w_u^J	w_f^J
1	50	20	1
2	10	1	50
3	1	0.2	50

The results of the five different parameterisation methods are presented in Fig. 7.11 to Fig. 7.13. In each parameterisation method, the 1) the planned trajectory using the given weights up-to the 30-th step, 2) the translation error $\epsilon^{\mathrm{t}}$and 3) the orientation error ϵ^{r} till step 30 are demonstrated which are computed as below

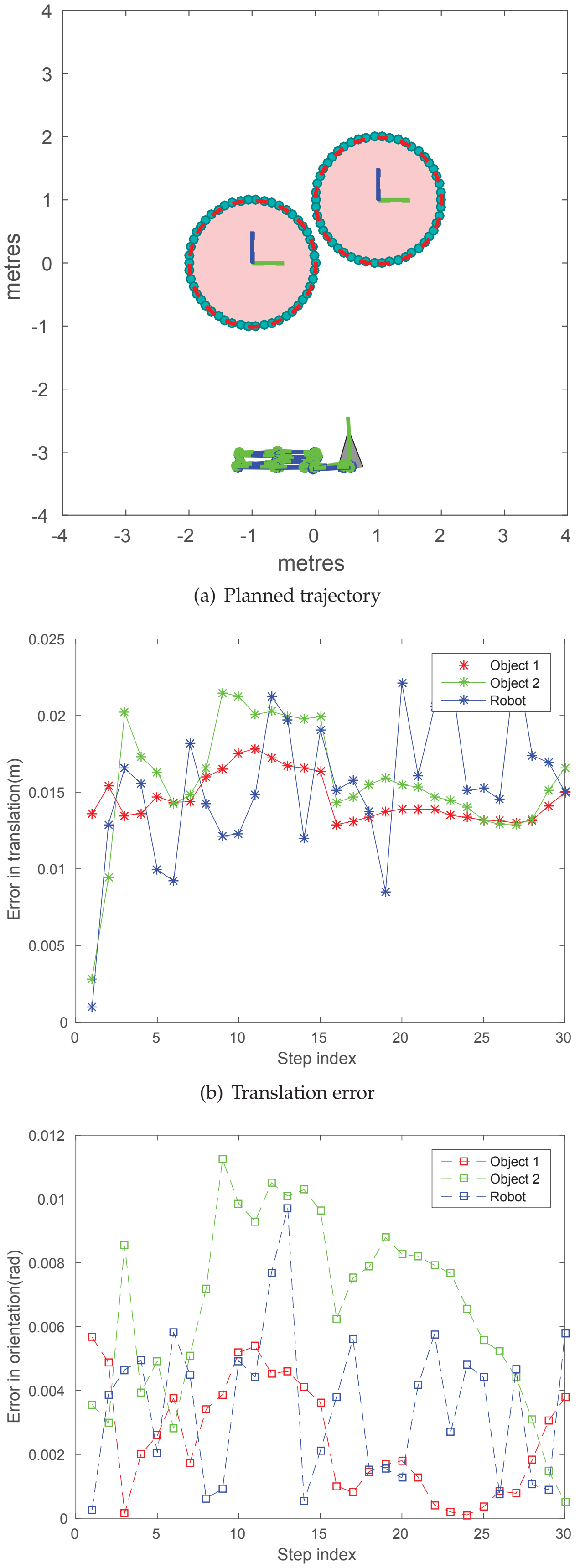
$$\begin{aligned} \epsilon^{\mathrm{t}} &= \sqrt{(x - \acute{x})^2 + (y - \acute{y})^2} \\ \epsilon^{\mathrm{r}} &= |\theta - \acute{\theta}^o| \end{aligned} \tag{7.53}$$

where the groundtruth pose is denoted as $\acute{\mathbf{p}} = \left[\acute{x}, \acute{y}, \acute{\theta}\right]$ and estimated pose is denoted as $\mathbf{p} = [x, y, \theta]$.

By examining and comparing the results of different combinations of w_X^J, w_u^J and w_f^J, the following conclusions are drawn:

1. As Fig. 7.11 demonstrates, relying on either uncertainty term $w_X^J \mathbf{c}_X \left(\mathbf{b}\left(\mathbf{X}_{k+L}^{\Delta}\right)\right)$ and control term $w_u^J \mathbf{c}_u \left(u_{l:k+L-1}\right)$ will not be able to generate reasonable planning trajectories and the robot is driven around a fixed position. Meanwhile, the estimation error is limited in a shorter range within 0.025m in translation and 0.012 rad which are significantly smaller compared to other 2 parameterisation methods by focusing on the state uncertainty term. Additionally, less control inputs will introduce less uncertainty and noises compared with larger control inputs thus leading to better estimation results. However, the robot is *not* able to cover features on the opposite side of the objects thus degenerating the

2. Fig. 7.12 shows good planning results which can cover most of the features on the objects. However, by looking at the figures in detail, as the dashed red lines on Fig. 7.12(a) indicate, the trajectory is composed from 3 line segments approximately. This is caused by the larger weight on the control term.

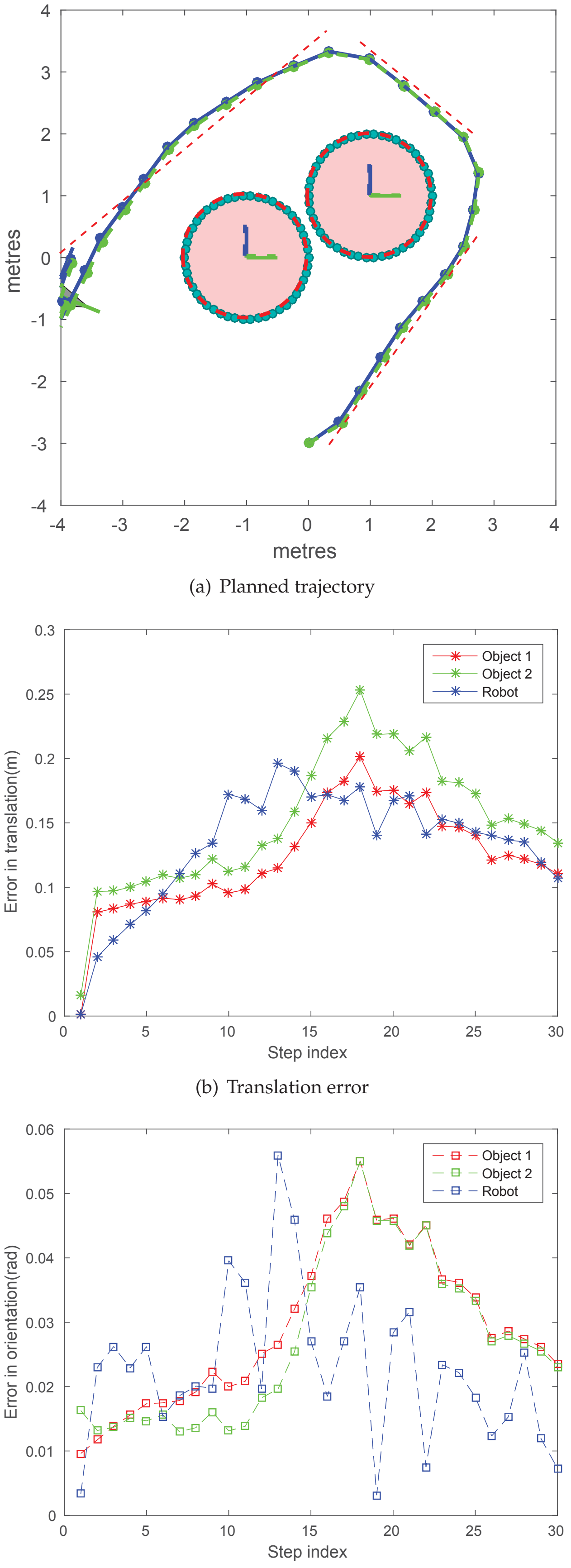


(a) Planned trajectory

(b) Translation error

(c) Orientation error

FIGURE 7.11: Experimental results of objective function parameterisation 1 in Table. 7.2.

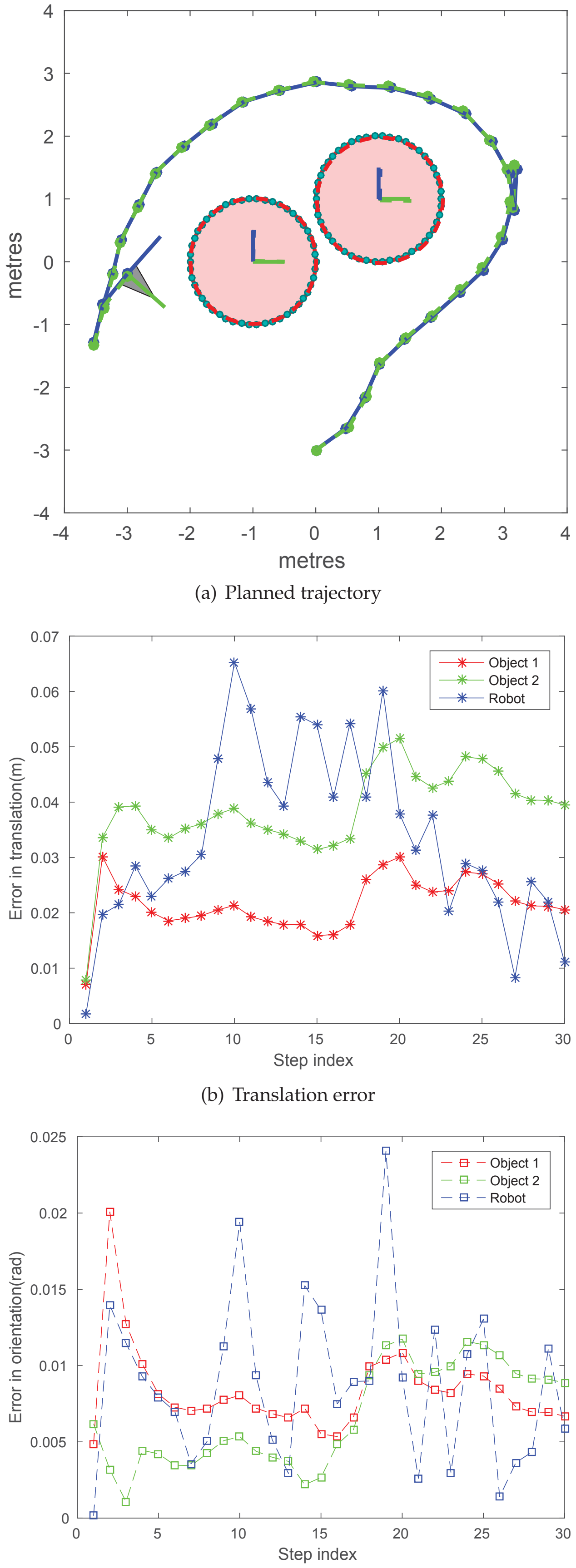


(a) Planned trajectory

(b) Translation error

(c) Orientation error

FIGURE 7.12: Experimental results of objective function parameterisation 2 in Table. 7.2.



(a) Planned trajectory

(b) Translation error

(c) Orientation error

FIGURE 7.13: Experimental results of objective function parameterisation 3 in Table. 7.2.

3. Fig. 7.13 shows a smoothed trajectory planned by 3rd combination of the weights. Under this parameterisation, as illustrated in previous sections, larger weight is assigned to the feature coverage term and the estimate uncertainty is the term with less priority. The control consumption term is assigned with the smallest weight.

After the empirical comparison in this subsection, we use the weights in parameters 3 in Table. 7.2 in the later simulation experiments.

7.4.4 Case Study 1: Trajectory Planning in Another Scenario

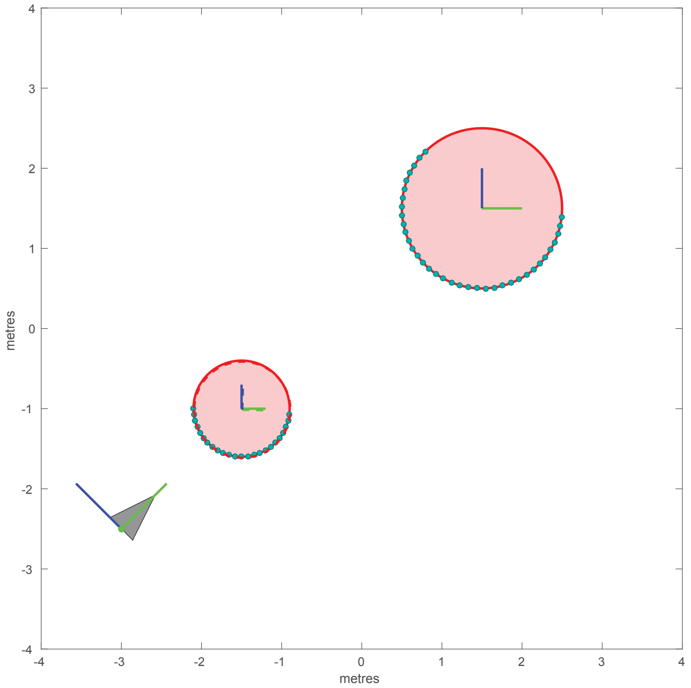


FIGURE 7.14: Simulation environment of scenario 2.

From Section. 7.4.2 and Section. 7.4.3, it has been shown that under environments of multiple objects, our framework is able to provide reasonable results. This section presents a more challenging case as shown in Fig. 7.14. There are two circular objects existing in the environment with different radius 0.6 m and 1 m located at $[-1.5, -1]$ and $[1.5, 1.5]$. For small object 1, 24 features are equally distributed from angle π to 2π and for larger object 2, 36 features are equally distributed between angle 0.75π to 2π. At the beginning step, the robot is not able to observe object 2 and hypothesis for object 2 has to be initialised during the planning phase after the features on object 2 are observed.

Fig. 7.15 shows the planned trajectory under this scenario. Before the red, dashed line on the trajectory, the robot covers most of the features on object 1. After acquiring features on object 2, the robot starts covering and re-observing the features on object 2 by planning to the left side of the object 2.

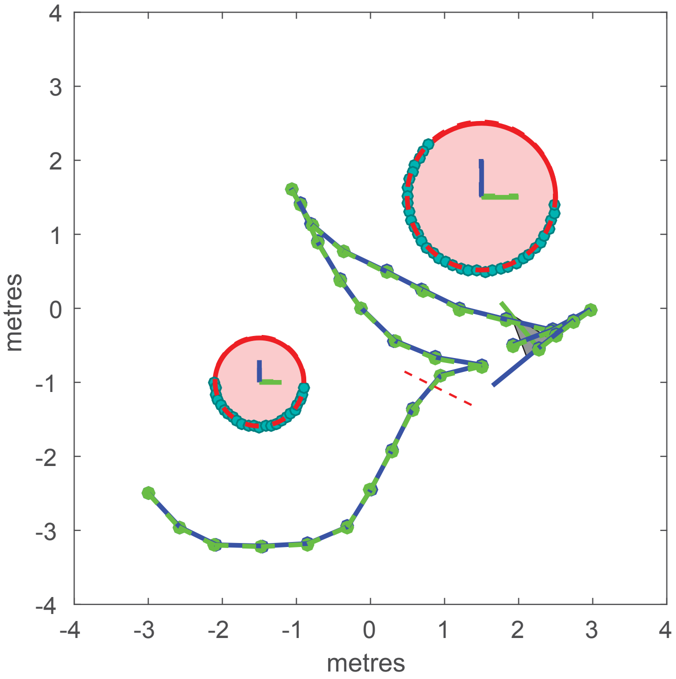
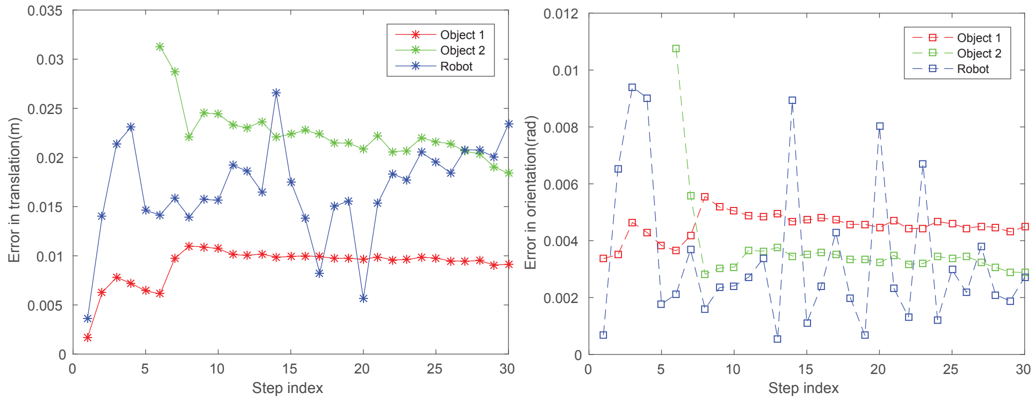


FIGURE 7.15: Planned trajectory in the scenario 2.

Fig. 7.16 shows the estimate errors from the planned trajectory. Object 2 is observed from step 5 with larger uncertainty which is significant from the initialisation of object 1. However, after observing more features from object 2, the estimate error is reduced gradually as expected.



(a) Translation error.

(b) Orientation error.

FIGURE 7.16: Estimation errors in the scenario 2.

7.4.5 Case Study 2: Obstacle Avoidance and Occlusion Handling

The above sections successfully demonstrate the effectiveness of the proposed framework under an environment which contains multiple objects, as shown in Fig. 7.8 and Fig. 7.14. With correct parameterisations, as was discussed above, the planned trajectory can cover most of the features on the objects while bounding the translation error and orientation error of the object within limited ranges. This section validates the approach under more complex scenarios which includes forbidden regions and occlusions.

At first, the simulation environment which only consists of a forbidden region drawn as the gray rectangle and target objects is presented as shown in Fig. 7.17. The groundtruth poses of two objects are $[-1, 0, 0]$ and $[1, 1, 0]$ and the radius of both objects are 0.6m. In the objective function, if the robot pose falls into the circular areas located along the border of the obstacle, a large penalty term will be added to the objective function, as illustrated in (7.46).

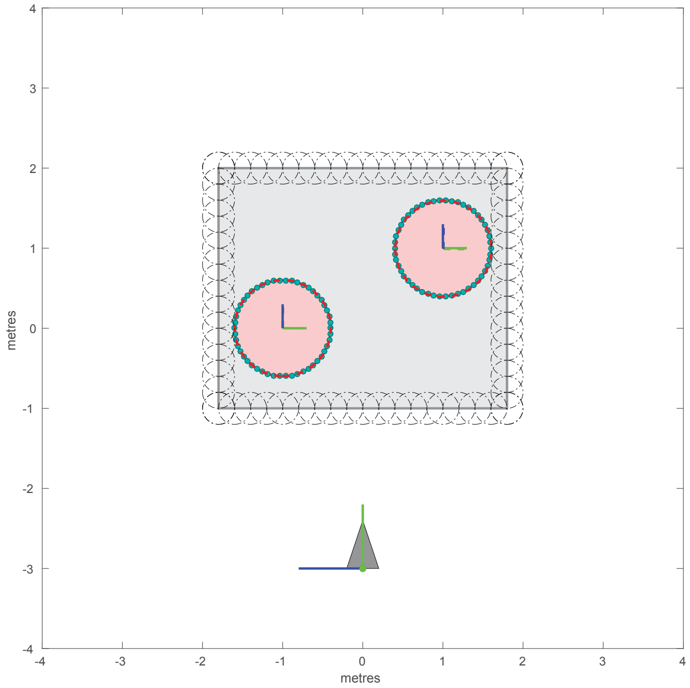


FIGURE 7.17: Environment with known obstacles.

Fig. 7.18 shows the planned trajectory. As the result indicates, the trajectory perfectly avoids the collision between the robot and the forbidden region and thus verifies the effectiveness of the additional penalty term. Meanwhile, Fig. 7.19 also demonstrates that the estimate errors are also limited within 0.12m and 0.03 rad.

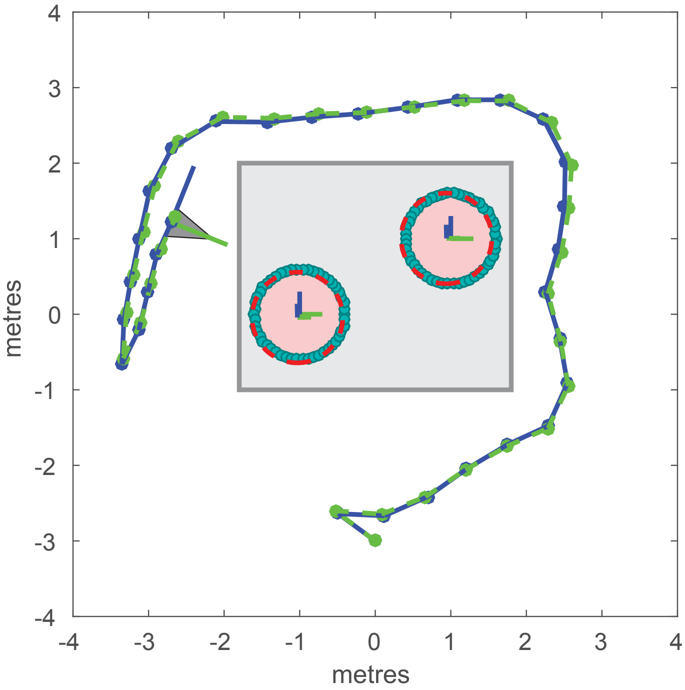
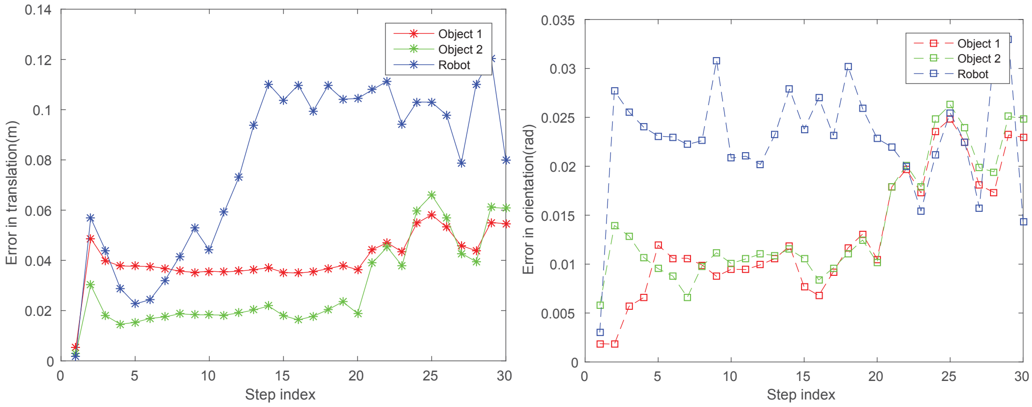


FIGURE 7.18: Planned trajectory in environment with known obstacles.

Except for the forbidden areas, in practical problems, obstacles may appear in the environment which may introduce not only collisions but also unknown occlusions. Here, a simulation environment in Fig. 7.20 is constructed where the gray rectangle denotes the forbidden region and the purple circle represents an obstacle which introduces problems of both collision and occlusions.

Following Section. 7.3.3, here it is assumed that there is a limited area behind the captured unknown area. Fig. 7.21 shows the understanding of the environment from the robot perspective at the beginning step. The red stars show the observed features on the objects and the colour-shaded lines on the purple circle denote the occlusion area given the current understanding of the environment. During the planning phase, the occlusion due to the colour-shaded area will reduce the number of observable features on the right region in the map. Please note that the unknown obstacle is *not* modelled and there is *no* mapping thread existed in this work, the current understanding of the unknown obstacles will *only* affect the prediction at the current step. In step k, the planning for steps $k+1$ to $k+L$ is only depending on the occlusion model which is captured at step k and the previous knowledge of the obstacle is ignored.

Fig. 7.22 shows the planned trajectory in the simulation environment in Fig. 7.20. Similar to Fig. 7.18, the trajectory avoids the possible obstacles; moreover, a significant difference is that, due to the existence of the unknown obstacle (purple circle), the robot exploits the left region first



(a) Translation error. (b) Orientation error.

FIGURE 7.19: Estimation errors in environment with a known obstacle.

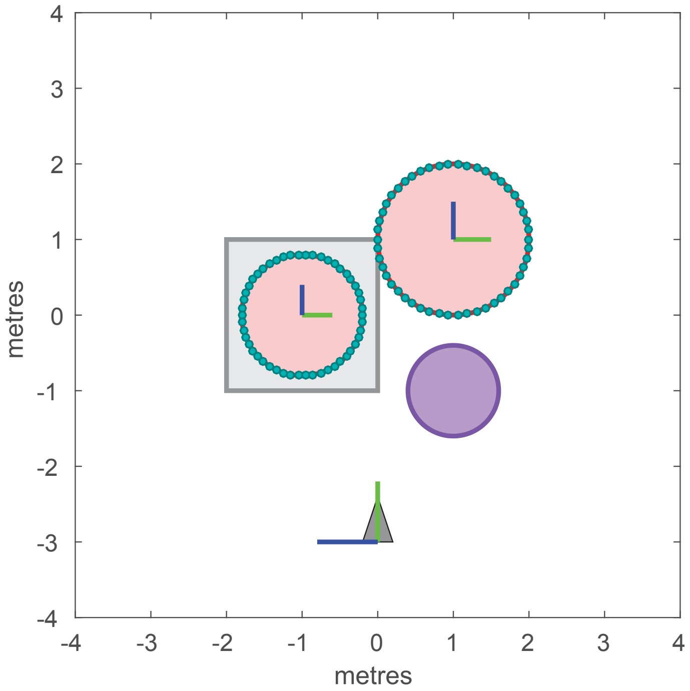


FIGURE 7.20: Environment with both known obstacles and unknown obstacles.

and then covers the right side. This representative trajectory demonstrates the effectiveness of our approach. In addition, the estimation errors are also presented in Fig. 7.23.

7.4.6 Case Study 3: Hypothesis Changing During Planning

This subsection validates the proposed strategy in the scenario where object hypothesis is changed during the planning phase which can happen when new features are observed, and the algorithm discovers that the previous understanding of the object is wrong.

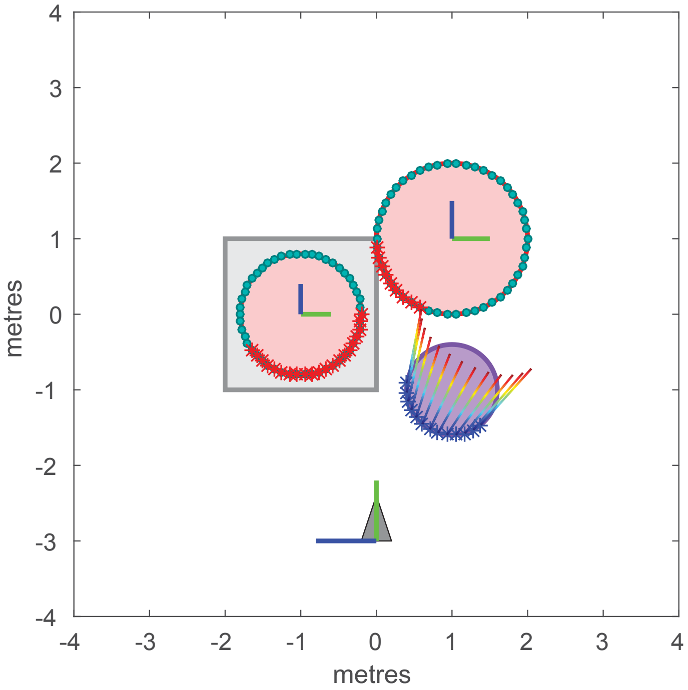


FIGURE 7.21: Environment understanding at the beginning step.

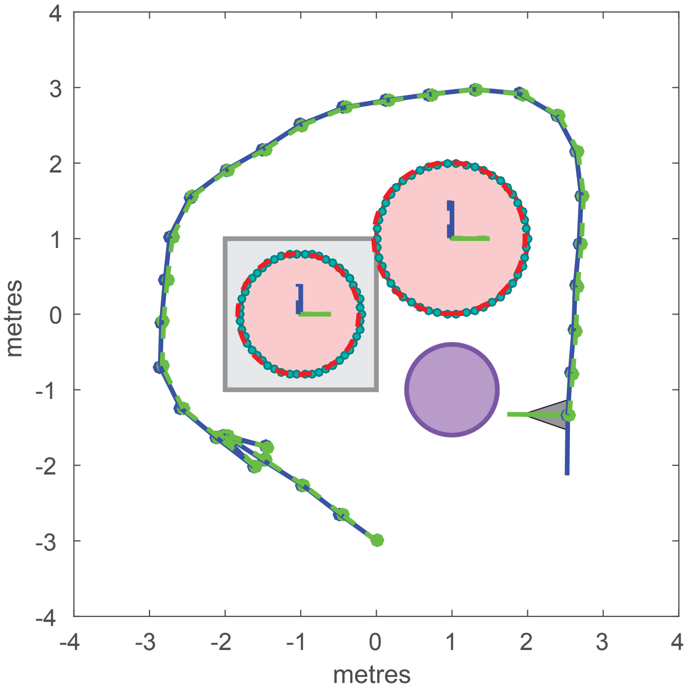
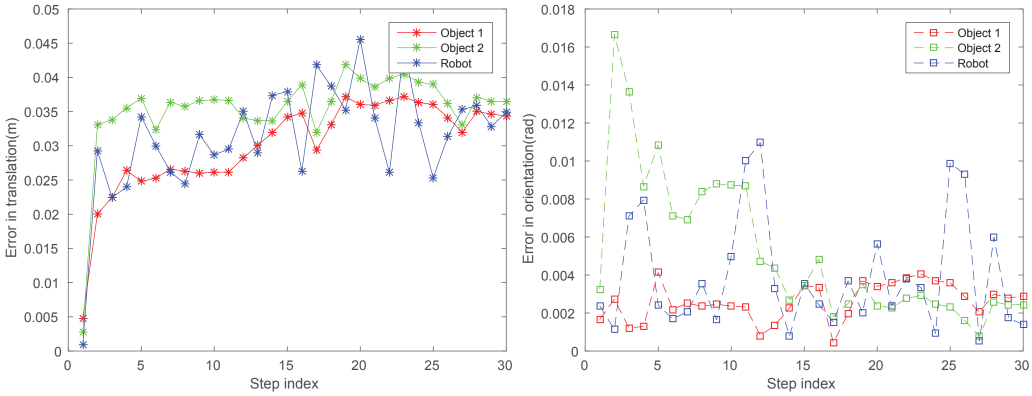


FIGURE 7.22: Planned trajectory in the environment with a forbidden region and an obstacle.

The simulation scenario shown in Fig. 7.24 is explained as follows: the radius of the object is equal to 1 m and located at $[0, 0, 0]$. The robot starts planning from pose $\left[0, -3, \frac{\pi}{2}\right]$. Object 1 has



(a) Translation error.

(b) Orientation error.

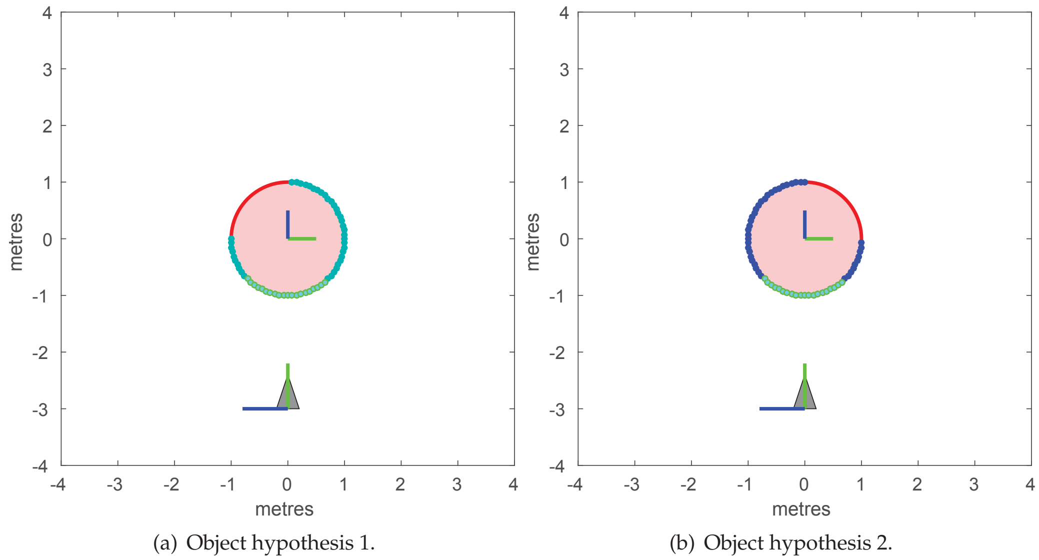
FIGURE 7.23: Estimation errors in the environment with a forbidden region and an obstacle.

60 features uniformly distributed between angle π to 2.5π and object 2 has 60 features uniformly distributed between angle 0.5π to 2π. The features between 1.25π and 1.75π are the same across two objects shown as the green dots, and the other dots denote the different features. At the beginning, the target object is recognised as object 1 and once the different features are associated, the robot discovers that the target object is object 2 rather than object 1. The trajectory is re-planned once the hypothesis is changed.

The planned trajectory is shown in Fig. 7.25. As the trajectory indicates, since the beginning assumption is object 1 and there are more features on the right side, therefore, the robot planned trajectory to the right side and executed. After the robot realises that the features on the right side are from object 2, the object hypothesis is corrected, shown as the dashed red rectangle in Fig. 7.25, the planned trajectory is replanned to the left side and the robot seeks more features from the left side to confirm object 2. Fig. 7.26 shows the translation error and orientation error of the robot and the object up to step 20.

7.5 Conclusion

This chapter presents an active object detection and pose estimation algorithm in general belief space. The proposed algorithm considers uncertainties introduced in both motion model and observation model, and formulates the planning problem as an optimal control problem using MPC. The objective function is designed delicately considering the feature coverage on the target objects, the estimation uncertainty of objects and robots and control consumption. The feature



(a) Object hypothesis 1.

(b) Object hypothesis 2.

FIGURE 7.24: Similar object with different distributed feature.

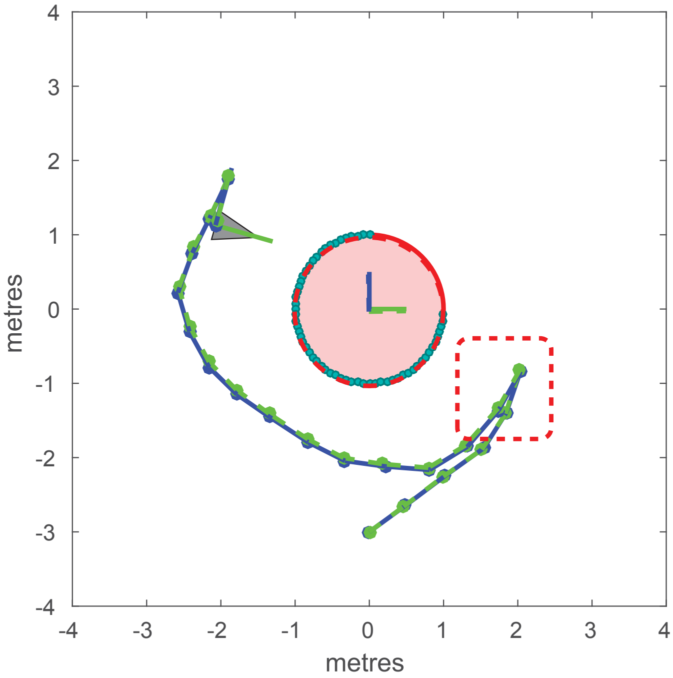
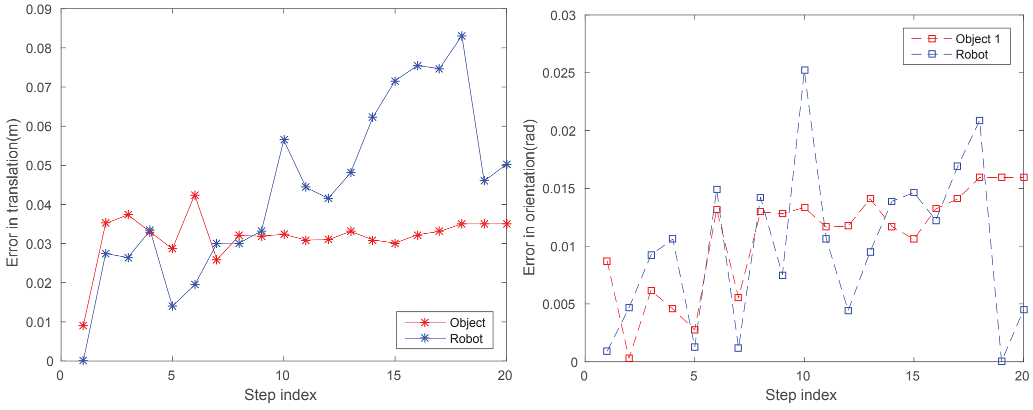


FIGURE 7.25: Planned trajectory in the environment where object hypothesis is changed after observing new features.

association probability is modelled for simplicity using a Gaussian distribution to predict the future observations. The proposed algorithm is able to re-plan the trajectory by computing the KL-divergence between the prior belief and posterior belief. Several practical problems are addressed such as object state (mean and covariance) initialisation, occlusion handling, collision avoidance



(a) Translation error.

(b) Orientation error.

FIGURE 7.26: Estimate error corresponding to the trajectory in Fig. 7.25.

and object hypothesis changing after updating from observations. Due to the lack of a mature experiment platform and insufficient time, practical experiments is not added. However, through comprehensive simulation experiments by taking different types of noises into consideration, the effectiveness of the proposed algorithm has been verified.

Chapter 8

Conclusion

This thesis investigates the problem of active object recognition and pose estimation. The traditional feature-based textured object detection and pose estimation methods are reviewed first. Strategies to improve the effectiveness and efficiency of these feature-based texured object detection and pose estimation are developed using a modern RGB-D sensor. To detect objects under severe illumination conditions, a novel RGB-D feature is proposed which is robust to illumination, viewpoint, scale and rotation variations and thus providing reliable feature matching results even for less-textured objects. The proposed methods are implemented and validated for a robotic perception module under a warehouse environment. Based on the proposed work in single-view object detection and pose estimation, two types of active object detection and pose estimation system are presented: 1) a model-driven next-best-view planning algorithm by exploiting the feature association probability which works effectively even with a naive greedy search planning method; 2) an optimisation-based framework which computes the trajectory for future steps considering the motion and observation uncertainties.

In this chapter, the contributions in this thesis are restated, and the limitations and future works are discussed.

8.1 Single-View Object Detection and Pose Estimation

Contributions

- In Chapter 3, a fast, robust and modular textured object detection and pose estimation framework under a cluttered indoor environment is proposed by taking the advantages of a novel RGB-D sensor specifically in feature correspondences clustering and pose estimation. An outlier rejection algorithm is also presented using a relational graph constructed from

3D-3D correspondences. Compared with the traditional RANSAC approach, the proposed graph-based approach takes less time while achieving robust outlier rejection performances. The framework also demonstrates reliable object detection and accurate pose estimation results.

- In Chapter 4, a novel RGB-D feature which is robust to illumination, viewpoint, scale and rotation variations is proposed. Significance of designing the keypoint detector and feature descriptor simultaneously when building the proposed RGB-D feature is highlighted. In RISAS, the geometric information provided from the depth channel of an RGB-D sensor is combined with texture information. RISAS shows superior performance compared with state-of-the-art 2D, 3D and RGB-D features, and is validated for object detection under severe illumination environments.
- By combining the work in Chapter 3 with a kernel descriptor, a practical, effective and efficient object detection and estimation framework under warehouse environments is presented. The target objects are categorised into different classes based on the rigidity, texture information and the type of its package/surface. This compound robotic perception system has been used by the collaborative team Z.U.N at the Amazon Picking Challenge 2015. This perception system is able to detect the target object and provides the grasping region of a candidate object.

Limitations and Future Work

The work in both Chapter. 3 and Chapter. 4 requires the target object to be information rich in either appearance or geometry where the discriminative features can be extracted and matched. Apart from this, the objects are required to be rigid. In comparison to the traditional feature-based object detection methods, the work presented in this thesis has achieved significant performance gains. However, deep learning based approaches achieve the state-of-the-art of object detection performances in category-level object recognition. The following two key aspects can be further investigated from the deep learning perspective

1. How to adopt state-of-the-art deep learning object detection approaches which is aimed at category-level object detection such as R-CNN[57], You Only Look Once (YOLO[120]) and Single Shot Multibox Detector (SSD[94]) into instance-level object detection problem for robotic perception?

2. 6 DoFs pose estimation can be formulated as a least square optimisation problem trivially using 2D-3D, 3D-3D or even 2D-2D correspondences in a feature-based object detection framework. However, how to integrate the pose estimation problem within a deep learning framework has not been explored as yet.

8.2 Active Object Detection and Pose Estimation

Contributions

- In Chapter. 6, a novel active object recognition and pose estimation system is introduced using two types of object models: 1) a sparse feature model, augmented with the characteristics of features when observed from different viewpoints and 2) a dense point cloud model which facilitates storing geometry. This dual model strategy makes it possible to accurately predict the expected information available during the Next-Best-View planning process as both the visibility as well as the likelihood of feature matching can be considered simultaneously. Another parameter to differentiate objects with similar appearances is attached to each feature which denotes its uniqueness across all modelled objects. The proposed strategy can identify the discriminative features of each object easily and guides the sensor to the viewpoints which can differentiate the target objects unambiguously. The effectiveness of the proposed active object detection and pose estimation framework is demonstrated using an RGB-D sensor.
- In order to incorporate motion and observation uncertainties into the active object detection and pose estimation, Chapter. 7 presents another active perception framework formulated using planning under uncertainty. By carefully designing the objective function considering the estimation uncertainty, feature coverage and control consumption, the proposed framework is able to achieve optimised control sequence for a desired trajectory. In Chapter. 7, various issues are addressed such as the object pose and covariance initialisation, initial guess for control optimisation, obstacle avoidance and occlusion handling and online re-planning. By conducting simulation experiments thoroughly, the effectiveness of the proposed framework is validated. This proposed framework has the potential to be extended for solving the active SLAM problem.

Limitations and Future Work

- In Chapter. 6, the key contribution is to model the feature association capability under different variations such as scale and viewpoint. This helps to lead to more accurate prediction in future observation under a selected viewpoint. It will be beneficial to model the impact of additional conditions such as illumination and rotation to enhance the accuracy of these predictions. We would also like to combine a more advanced planner with the current framework and validate its effectiveness. Lastly, in recent years the conventional feature-based object recognition methods such as [31] have been surpassed by superior model-based approaches using more advanced computer vision techniques such as deep learning. In recent work[111, 11], a model which describes the object detection confidence under different viewpoints is built by collecting a large amount of data with careful training. How to integrate such approaches with strategies developed in this thesis will be a beneficial direction.

- In the planning framework presented in Chapter. 7, it is possible to add a complementary mapping thread which can work in parallel with the current framework. In the proposed system, even though the obstacles can be handled correctly, they are not modelled explicitly, and *only* the current understanding of the obstacles is used to predict the future observation. However, modelling the obstacle will provide more information about the environment and enable more accurate prediction of the future observation thus planning a better robot trajectory. From a theoretical perspective, current probabilistic feature association modelling can be further improved by training with more data. Besides, work has already been started to replace the current state representation using Lie group which enables more accurate state propagation. Implementing active object detection and pose estimation strategies on a mobile manipulator such as Fetch robot is an important avenue to demonstrate the effectiveness of the proposed algorithms.

- In Part 2 of the thesis, some of the contributions presented in Part 1 are not implemented. The proposed RGB-D feature has not been validated in active object detection and pose estimation frameworks even though there is no difficulties to integrate RISAS into proposed algorithms theoretically. However, this remains to be a future work of the thesis. On the other hand, it also will be beneficial to improve the object detection and pose estimation performance in warehouse environments if active perception strategies are employed.

Strengthening the connection between single view perception algorithms and active perception algorithms introduced in this thesis will be the key goal of the future work.

Appendix A

Proofs in Chapter. 7

A.1 Linearisation of (7.14)

In the end of (7.14), we obtain:

$$\begin{aligned}&\underset{\mathbf{X}_{k+l}}{\arg\min} \left\|\mathbf{X}_k - \mathbf{X}_k^{\Delta}\right\|^2_{\mathbf{I}_k} + \sum_{i=1}^{l} \left\|\mathbf{X}_{k+i} - g\left(\mathbf{X}_{k+i-1}, u_{k+i-1}\right)\right\|^2_{\mathbf{I}_\eta} + \\ &\sum_{i=1}^{l}\sum_{j=1}^{n_i} \left\|\mathbf{z}_{k+i,j} - h\left(\mathbf{X}_{k+i}, f_j^{\mathrm{o}}\right)\right\|^2_{\bar{\mathbf{I}}_\xi^{i,j}}\end{aligned} \tag{A.1}$$

Now the following shows the conversion from (7.14) to error state representation (7.15). At time step k, it is trivial to have:

$$\|\mathbf{X}_k - \mathbf{X}_k^{\Delta}\|^2_{\mathbf{I}_k} = \|\Delta\mathbf{X}_k\|^2_{\mathbf{I}_k} \tag{A.2}$$

For the second term:

$$
\begin{aligned}
&\sum_{i=1}^{l} \|\mathbf{X}_{k+i} - \mathbf{F}\left(\mathbf{X}_{k+i-1}, u_{k+i-1}\right)\|_{\mathbf{I}_\eta}^2 \\
=&\sum_{i=1}^{l} \left\|\mathbf{X}_{k+i} - \bar{\mathbf{X}}_{k+i} + \bar{\mathbf{X}}_{k+i} - \mathbf{F}\left(\mathbf{X}_{k+i-1}, u_{k+i-1}\right)\right\|_{\mathbf{I}_\eta}^2 \\
=&\sum_{i=1}^{l} \left\|\Delta\mathbf{X}_{k+i} + \bar{\mathbf{X}}_{k+i} - \mathbf{F}\left(\mathbf{X}_{k+i-1}, u_{k+i-1}\right)\right\|_{\mathbf{I}_\eta}^2 \\
=&\sum_{i=1}^{l} \left\|\Delta\mathbf{X}_{k+i} + \mathbf{F}\left(\bar{\mathbf{X}}_{k+i-1}, u_{k+i-1}\right) - \mathbf{F}\left(\mathbf{X}_{k+i-1}, u_{k+i-1}\right) + \bar{\mathbf{X}}_{k+i} - \mathbf{F}\left(\bar{\mathbf{X}}_{k+i-1}, u_{k+i-1}\right)\right\|_{\mathbf{I}_\eta}^2 \\
=&\sum_{i=1}^{l} \left\|\Delta\mathbf{X}_{k+i} - \frac{\partial\mathbf{F}}{\partial\mathbf{X}_{k+i-1}}\Delta\mathbf{X}_{k+i-1} + \bar{\mathbf{X}}_{k+i} - \mathbf{F}\left(\bar{\mathbf{X}}_{k+i-1}, u_{k+i-1}\right)\right\|_{\mathbf{I}_\eta}^2 \\
=&\sum_{i=1}^{l} ||\Delta\mathbf{X}_{k+i} - \frac{\partial\mathbf{F}}{\partial\mathbf{X}_{k+i-1}}\Delta\mathbf{X}_{k+i-1}||_{\bar{\mathbf{I}}_\eta}^2
\end{aligned}
\tag{A.3}
$$

where $\bar{\mathbf{X}}_{k+i} = \mathbf{F}\left(\bar{\mathbf{X}}_{k+i-1}, u_{k+i-1}\right)$ according to 7.16. The similar method is adopted for the last term:

$$
\begin{aligned}
&\sum_{i=1}^{l}\sum_{j=1}^{n_i} \left\|\mathbf{z}_{k+i,j} - h\left(\mathbf{X}_{k+i}, f_j^{\mathrm{o}}\right)\right\|_{\bar{\mathbf{I}}_\xi^{i,j}}^2 \\
=&\sum_{i=1}^{l}\sum_{j=1}^{n_i} \left\|\mathbf{z}_{k+i,j} - h\left(\bar{\mathbf{X}}_{k+i}, f_j^{\mathrm{o}}\right) + h\left(\bar{\mathbf{X}}_{k+i}, f_j^{\mathrm{o}}\right) - h\left(\mathbf{X}_{k+i}, f_j^{\mathrm{o}}\right)\right\|_{\bar{\mathbf{I}}_\xi^{i,j}}^2 \\
=&\sum_{i=1}^{l}\sum_{j=1}^{n_i} \left\|\frac{\partial h_j}{\partial\mathbf{X}_{k+i}}\Delta\mathbf{X}_{k+i} - b_{i,j}^h(\mathbf{z}_{k+i,j})\right\|_{\bar{\mathbf{I}}_\xi^{ij}}^2
\end{aligned}
\tag{A.4}
$$

where $b_{i,j}^h\left(\mathbf{z}_{k+i,j}\right) = h\left(\mathbf{X}_{k+i}, f_j^{\mathrm{o}}\right) - h\left(\bar{\mathbf{X}}_{k+i}, f_j^{\mathrm{o}}\right)$

A.2 Quadratic Form Representation

In order to obtain the quadratic representation from (7.15), the following two equation need to used

$$\begin{aligned}\text{Given } \mathbf{x},\mathbf{y} \quad ||\mathbf{x}||^2_{\mathbf{I}_x} + ||\mathbf{y}||^2_{\mathbf{I}_y} &= \left\| \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \right\|^2_{\begin{pmatrix} \mathbf{I}_x & 0 \\ 0 & \mathbf{I}_y \end{pmatrix}} \\ \sum_j^n \|\mathbf{A}_j\mathbf{x} - \mathbf{b}_j\|^2_{\mathbf{I}_j} &= \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2_{\begin{pmatrix} \mathbf{I}_1 & & \\ & \ddots & \\ & & \mathbf{I}_n \end{pmatrix}}\end{aligned} \tag{A.5}$$

Starting from the first term in (7.15)

$$\begin{aligned}\|\Delta\mathbf{X}_k\|^2_{\mathbf{I}_k} &= (\Delta\mathbf{X}_k)^\intercal \mathbf{I}_k \Delta\mathbf{X}_k \\ &= (\Delta\mathbf{X}_k)^\intercal \mathbf{I}_k^{1/2}\mathbf{I}_k^{1/2}\Delta\mathbf{X}_k \\ &= (\mathbf{I}_k^{1/2}\Delta\mathbf{X}_k)^2\end{aligned}$$

For the second term, the motion model related term, we have:

$$\begin{aligned}\sum_{i=1}^{l} \|\Delta\mathbf{X}_{k+i} - \mathbf{F}'\Delta\mathbf{X}_{k+i-1}\|^2_{\mathbf{I}_\eta} &= \sum_{i=1}^{l} \left\| \mathbf{I}_\eta^{1/2}\Delta\mathbf{X}_{k+i} - \mathbf{I}_\eta^{1/2}\mathbf{F}'\Delta\mathbf{X}_{k+i-1} \right\| \\ &= \mathcal{G}\Delta\mathbf{X}\end{aligned}$$

where $\mathbf{F}'_i$ is the short for $\frac{\partial\mathbf{F}}{\partial\mathbf{X}_{k+i-1}}$ and $\mathcal{G}$ is

$$\mathcal{G} = \left[\begin{array}{cccc|ccccc} 0 & \cdots & \cdots & -\mathbf{I}_\eta^{1/2}\mathbf{F}_1 & \mathbf{I}_\eta^{1/2} & 0 & \cdots & 0 & 0 \\ 0 & \cdots & \cdots & 0 & -\mathbf{I}_\eta^{1/2}\mathbf{F}_2 & \mathbf{I}_\eta^{1/2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & 0 & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & -I_\eta^{1/2}\mathbf{F}_{l-1} & \mathbf{I}_\eta^{1/2} \end{array}\right] \tag{A.6}$$

and $\Delta\mathbf{X}$ is

$$
\begin{aligned}
&\sum_{i=1}^{l}\sum_{j=1}^{n_i}\left\|h'_j\Delta\mathbf{X}_{k+i} - b^h_{i,j}(\mathbf{z}_{k+i,j})\right\|^2_{\bar{\mathbf{I}}^{ij}_{\xi}} \\
=&\sum_{i=1}^{l}\sum_{j=1}^{n_i}\left\|(\bar{\mathbf{I}}^{ij}_{\xi})^{1/2}h'_j\Delta\mathbf{X}_{k+i} - (\bar{\mathbf{I}}^{ij}_{\xi})^{1/2}b^h_{i,j}(\mathbf{z}_{k+i,j})\right\|^2 \\
=&\sum_{i=1}^{l}\left\|\tilde{\mathcal{H}}_i\Delta\mathbf{X}_{k+i} - \beta^h_i\right\|^2
\end{aligned}
\tag{A.7}
$$

where

$$
\tilde{\mathcal{H}}_i = \begin{bmatrix} \left(\bar{\mathbf{I}}^{i,1}_{\xi}\right)^{1/2} h'_1 \\ \left(\bar{\mathbf{I}}^{i,2}_{\xi}\right)^{1/2} h'_2 \\ \vdots \\ \left(\bar{\mathbf{I}}^{i,n_i}_{\xi}\right)^{1/2} h'_{n_i} \end{bmatrix} \text{ and } \beta^h_i = \begin{bmatrix} \left(\bar{\mathbf{I}}^{i,1}_{\xi}\right)^{1/2} b^h_{i,1}(\mathbf{z}_{k+i,1}) \\ \left(\bar{\mathbf{I}}^{i,2}_{\xi}\right)^{1/2} b^h_{i,2}(\mathbf{z}_{k+i,2}) \\ \vdots \\ \left(\bar{\mathbf{I}}^{i,n_i}_{\xi}\right)^{1/2} b^h_{i,n_i}(\mathbf{z}_{k+i,n_i}) \end{bmatrix}
\tag{A.8}
$$

From eq. A.5, we can stack $\tilde{\mathcal{H}}_i$ and β^h_i from step $k+1$ to step $k+l$ to obtain $\tilde{\mathcal{H}}$ and β^h. Finally, in quadratic form eq. 7.17, we have:

$$
\mathcal{A}_{k+l} = \begin{bmatrix} \begin{bmatrix} \mathbf{I}^{1/2}_k & \mathbf{0} \end{bmatrix} \\ \mathcal{G} \\ \tilde{\mathcal{H}} \end{bmatrix} \text{ and } \mathcal{B}_{k+l} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \beta^h \end{bmatrix}
\tag{A.9}
$$

Bibliography

[1] Mongi A. Abidi and T Chandra. “A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation”. In: *IEEE transactions on pattern analysis and machine intelligence* 17.5 (1995), pp. 534–538.

[2] Ali-akbar Agha-mohammadi, Suman Chakravorty, and Nancy M Amato. “FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements”. In: *The International Journal of Robotics Research* 33.2 (2014), pp. 268–304.

[3] Aitor Aldoma et al. “A global hypotheses verification method for 3d object recognition”. In: *European Conference on Computer Vision*. Springer. 2012, pp. 511–524.

[4] Aitor Aldoma et al. “CAD-model recognition and 6DOF pose estimation using 3D cues”. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE. 2011, pp. 585–592.

[5] Aitor Aldoma et al. “Point cloud library”. In: *IEEE Robotics & Automation Magazine* 1070.9932/12 (2012).

[6] Aitor Aldoma et al. “Tutorial: Point Cloud Library: Three-Dimensional Object Recognition and 6 DOF Pose Estimation”. In: *IEEE Robotics & Automation Magazine* 3.19 (2012), pp. 80–91.

[7] Marc-André Ameller, Bill Triggs, and Long Quan. “Camera Pose Revisited–New Linear Algorithms”. In: (2000).

[8] K. S. Arun, T. S. Huang, and S. D. Blostein. “Least-Squares Fitting of Two 3-D Point Sets”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9.5 (1987), pp. 698–700. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1987.4767965.

[9] K Somani Arun, Thomas S Huang, and Steven D Blostein. “Least-squares fitting of two 3-D point sets”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 5 (1987), pp. 698–700.

[10] Sunil Arya et al. "An optimal algorithm for approximate nearest neighbor searching fixed dimensions". In: *Journal of the ACM (JACM)* 45.6 (1998), pp. 891–923.

[11] Nikolay Atanasov et al. "Nonmyopic view planning for active object classification and pose estimation". In: *IEEE Transactions on Robotics* 30.5 (2014), pp. 1078–1090.

[12] Abraham Bachrach et al. "Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments". In: *The International Journal of Robotics Research* 31.11 (2012), pp. 1320–1343.

[13] Haoyu Bai, David Hsu, and Wee Sun Lee. "Integrated perception and planning in the continuous space: A POMDP approach". In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1288–1302.

[14] Ruzena Bajcsy. "Active perception". In: *Proceedings of the IEEE* 76.8 (1988), pp. 966–1005.

[15] Dana H Ballard. "Generalizing the Hough transform to detect arbitrary shapes". In: *Pattern recognition* 13.2 (1981), pp. 111–122.

[16] Kurt Konolige Bastian Steder Radu Bogdan Rusu and Wolfram Burgard. "NARF: 3D Range Image Features for Object Recognition". In: *Proc. Workshop on Defining and SOlving Realistic Perception Problems at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2010)*. 2010.

[17] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features". In: *Proc. European Conference on Computer Vision (ECCV' 2006)*. Vol. 3951. 2006, pp. 404–417.

[18] Herbert Bay et al. "Speeded-Up Robust Features (SURF)". In: *Computer Vision and Image Understanding* 110.3 (2008), pp. 346–359.

[19] P. R. Beaudet. "Rotationally invariant image operators". In: *International Conference on Pattern Recognition*. 1978.

[20] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection". In: *IEEE Transactions on pattern analysis and machine intelligence* 19.7 (1997), pp. 711–720.

[21] Ian Goodfellow Yoshua Bengio and Aaron Courville. "Deep Learning". Book in preparation for MIT Press. 2016. URL: `http://www.deeplearningbook.org`.

[22] Yoshua Bengio, Ian J Goodfellow, and Aaron Courville. "Deep learning". In: (2015).

[23] Andreas Bircher et al. “Receding horizon "next-best-view" planner for 3D exploration”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 1462–1468.

[24] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. “Kernel descriptors for visual recognition”. In: *Advances in neural information processing systems*. 2010, pp. 244–252.

[25] Ali Borji et al. “Salient object detection: A survey”. In: *arXiv preprint arXiv:1411.5878* (2014).

[26] Hermann Borotschnig et al. “Appearance-based active object recognition”. In: *Image and Vision Computing* 18.9 (2000), pp. 715–727.

[27] Björn Browatzki et al. “Active object recognition on a humanoid robot”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 2021–2028.

[28] Adam Bry and Nicholas Roy. “Rapidly-exploring random belief trees for motion planning under uncertainty”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 723–730.

[29] Michael Calonder et al. “Brief: Binary robust independent elementary features”. In: *Proc. European Conference on Computer Vision (ECCV’2010)* (2010), pp. 778–792.

[30] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. “Active vision in robotic systems: A survey of recent developments”. In: *The International Journal of Robotics Research* 30.11 (2011), pp. 1343–1377.

[31] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. “The MOPED framework: Object recognition and pose estimation for manipulation”. In: *The International Journal of Robotics Research* 30.10 (2011), pp. 1284–1306. DOI: `10.1177/0278364911401765`. eprint: `http://ijr.sagepub.com/content/30/10/1284.full.pdf+html`. URL: `http://ijr.sagepub.com/content/30/10/1284.abstract`.

[32] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. “The MOPED Framework: Object Recognition and Pose Estimation for Manipulation”. In: *The International Journal of Robotics Research* 30.10 (2011), pp. 1284–1306.

[33] Dorin Comaniciu and Peter Meer. “Mean shift: A robust approach toward feature space analysis”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.5 (2002), pp. 603–619.

[34] Cl Connolly. “The determination of next best views”. In: *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*. Vol. 2. IEEE. 1985, pp. 432–435.

[35] G. Costante et al. “Perception-aware Path Planning”. In: *ArXiv e-prints* (May 2016). arXiv: `1605.04151 [cs.RO]`.

[36] Gabriella Csurka et al. “Visual categorization with bags of keypoints”. In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 1-22. Prague. 2004, pp. 1–2.

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

[38] Joachim Denzler and Christopher M Brown. “Information theoretic sensor data selection for active object recognition and state estimation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2 (2002), pp. 145–157.

[39] M. W. M. G. Dissanayake et al. “A solution to the simultaneous localization and map building (SLAM) problem”. In: *IEEE Transactions on Robotics and Automation* 17.3 (2001), pp. 229–241. DOI: `10.1109/70.938381`.

[40] MWM Gamini Dissanayake et al. “A solution to the simultaneous localization and map building (SLAM) problem”. In: *IEEE Transactions on robotics and automation* 17.3 (2001), pp. 229–241.

[41] Alexander Domahidi. “FORCES: Fast optimization for real-time control on embedded systems”. In: *Available at forces. ethz* (2012).

[42] Andreas Doumanoglou et al. “6D Object Detection and Next-Best-View Prediction in the Crowd”. In: *arXiv preprint arXiv:1512.07506* (2015).

[43] Bertram Drost et al. “Model globally, match locally: Efficient and robust 3D object recognition.” In: *CVPR*. Vol. 1. 2. 2010, p. 5.

[44] David W Eggert, Adele Lorusso, and Robert B Fisher. “Estimating 3-D rigid body transformations: a comparison of four major algorithms”. In: *Machine Vision and Applications* 9.5-6 (1997), pp. 272–290.

[45] D.W. Eggert, A. Lorusso, and R.B. Fisher. “Estimating 3-D rigid body transformations: a comparison of four major algorithms”. In: *Machine Vision and Applications* 9.5 (1997), pp. 272–290. ISSN: 1432-1769. DOI: `10.1007/s001380050048`.

[46] F. Endres et al. “3-D Mapping With an RGB-D Camera”. In: *Robotics, IEEE Transactions on* 30.1 (2014), pp. 177–187.

[47] Olof Enqvist, Klas Josephson, and Fredrik Kahl. “Optimal correspondences from pairwise constraints”. In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 1295–1302.

[48] T Erez and WD Smart. “A Scalable Method for Solving High-Dimensional Continuous POMDPs Using Local Approximation”. In: *Proceedings of the 26th Conference in Uncertainty in Artificial Intelligence (UAI)*. 2010.

[49] Mark Everingham et al. “The pascal visual object classes challenge: A retrospective”. In: *International Journal of Computer Vision* 111.1 (2015), pp. 98–136.

[50] Pedro F Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2010), pp. 1627–1645.

[51] Guanghua Feng, Yong Liu, and Yiyi Liao. “LOIND: An illumination and scale invariant RGB-D descriptor”. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA' 2015)*. 2015, pp. 1893–1898.

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

[53] Torea Foissotte et al. “A two-steps next-best-view algorithm for autonomous 3d object modeling by a humanoid robot”. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 1159–1164.

[54] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. “An algorithm for finding best matches in logarithmic expected time”. In: *ACM Transactions on Mathematical Software (TOMS)* 3.3 (1977), pp. 209–226.

[55] Aristides Gionis et al. “Similarity search in high dimensions via hashing”. In:

[56] Ross Girshick. “Fast R-CNN”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2015, pp. 1440–1448.

[57] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[58] Kristen Grauman and Bastian Leibe. "Visual object recognition". In: *Synthesis lectures on artificial intelligence and machine learning* 5.2 (2011), pp. 1–181.

[59] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Siam, 2008.

[60] Yulan Guo et al. "A comprehensive performance evaluation of 3D local feature descriptors". In: *International Journal of Computer Vision* 116.1 (2016), pp. 66–89.

[61] Wee Sun Lee Hanna Kurniawati David Hsu. "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces". In: *Proceedings of Robotics: Science and Systems IV*. Zurich, Switzerland, 2008. DOI: 10.15607/RSS.2008.IV.009.

[62] Chris Harris and Mike Stephens. "A combined corner and edge detector." In: Citeseer. 1988.

[63] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[64] Ruijie He, Sam Prentice, and Nicholas Roy. "Planning in information space for a quadrotor helicopter in a GPS-denied environment". In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE. 2008, pp. 1814–1820.

[65] Stefan Hinterstoisser et al. "Gradient response maps for real-time detection of textureless objects". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (2012), pp. 876–888.

[66] Berthold KP Horn. "Closed-form solution of absolute orientation using unit quaternions". In: *JOSA A* 4.4 (1987), pp. 629–642.

[67] Berthold KP Horn, Hugh M Hilden, and Shahriar Negahdaripour. "Closed-form solution of absolute orientation using orthonormal matrices". In: *JOSA A* 5.7 (1988), pp. 1127–1135.

[68] Jan Hosang et al. "What makes for effective detection proposals?" In: *IEEE transactions on pattern analysis and machine intelligence* 38.4 (2016), pp. 814–830.

[69] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. “Comparing images using the Hausdorff distance”. In: *IEEE Transactions on pattern analysis and machine intelligence* 15.9 (1993), pp. 850–863.

[70] Vadim Indelman, Luca Carlone, and Frank Dellaert. “Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments”. In: *The International Journal of Robotics Research* 34.7 (2015), pp. 849–882.

[71] Piotr Indyk and Rajeev Motwani. “Approximate nearest neighbors: towards removing the curse of dimensionality”. In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM. 1998, pp. 604–613.

[72] Yangqing Jia and Trevor Darrell. “Heavy-tailed distances for gradient based image descriptors”. In: *Advances in Neural Information Processing Systems*. 2011, pp. 397–405.

[73] Andrew E. Johnson and Martial Hebert. “Using spin images for efficient object recognition in cluttered 3D scenes”. In: *IEEE Transactions on pattern analysis and machine intelligence* 21.5 (1999), pp. 433–449.

[74] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. “Planning and acting in partially observable stochastic domains”. In: *Artificial intelligence* 101.1 (1998), pp. 99–134.

[75] Sertac Karaman and Emilio Frazzoli. “Sampling-based algorithms for optimal motion planning”. In: *The International Journal of Robotics Research* 30.7 (2011), pp. 846–894.

[76] Lydia E Kavraki et al. “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”. In: *IEEE transactions on Robotics and Automation* 12.4 (1996), pp. 566–580.

[77] Yan Ke and Rahul Sukthankar. “PCA-SIFT: A more distinctive representation for local image descriptors”. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2004, pp. II–506.

[78] Wadim Kehl et al. “Deep Learning of Local RGB-D Patches for 3D Object Detection and 6D Pose Estimation”. In: *arXiv preprint arXiv:1607.06038* (2016).

[79] Christian Kerl, Jürgen Sturm, and Daniel Cremers. “Dense visual SLAM for RGB-D cameras”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 2100–2106.

[80] Kenji Kira and Larry A Rendell. “The feature selection problem: Traditional methods and a new algorithm”. In: *AAAI*. Vol. 2. 1992, pp. 129–134.

[81] Michael Krainin, Brian Curless, and Dieter Fox. “Autonomous generation of complete 3D object models using next best view manipulation planning”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 5031–5037.

[82] S. Kriegel et al. “Combining object modeling and recognition for active scene exploration”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS'2013)*. Nov. 2013, pp. 2384–2391.

[83] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[84] Neeraj Kumar, Li Zhang, and Shree Nayar. “What is a good nearest neighbors algorithm for finding similar patches in images?” In: *European conference on computer vision*. Springer. 2008, pp. 364–378.

[85] K. Lai et al. “Sparse distance learning for object recognition combining RGB and depth information”. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA'2011)*. 2011, pp. 4007–4013.

[86] Florent Lamiraux, David Bonnafous, and Olivier Lefebvre. “Reactive path deformation for nonholonomic mobile robots”. In: *IEEE Transactions on Robotics* 20.6 (2004), pp. 967–977.

[87] Steven M LaValle. “Rapidly-exploring random trees: A new tool for path planning”. In: (1998).

[88] Yann LeCun. “Deep learning tutorial”. In: Citeseer.

[89] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444.

[90] Honglak Lee et al. “Efficient sparse coding algorithms”. In: *Advances in neural information processing systems*. 2006, pp. 801–808.

[91] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. “Epnp: An accurate o (n) solution to the pnp problem”. In: *International journal of computer vision* 81.2 (2009), pp. 155–166.

[92] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. "BRISK: Binary robust invariant scalable keypoints". In: *2011 International conference on computer vision*. IEEE. 2011, pp. 2548–2555.

[93] Hongdong Li and Richard Hartley. "The 3D-3D registration problem revisited". In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE. 2007, pp. 1–8.

[94] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *arXiv preprint arXiv:1512.02325* (2015).

[95] David Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[96] David G Lowe. "Object recognition from local scale-invariant features". In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[97] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. "Ensemble of exemplar-svms for object detection and beyond". In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 89–96.

[98] Eric Marchand and François Chaumette. "Active vision for complete scene reconstruction and exploration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.1 (1999), pp. 65–72.

[99] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes". In: *International Journal of Computer Vision* 89.2-3 (2010), pp. 348–361.

[100] Krystian Mikolajczyk and Cordelia Schmid. "A performance evaluation of local descriptors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.10 (2005), pp. 1615–1630.

[101] Thomas Minka. "Expectation-Maximization as lower bound maximization". In: (1998).

[102] David M. Mount and Sunil Arya. *ANN: a library for approximate nearest neighbour searching*. `https://www.cs.umd.edu/~mount/ANN/`.

[103] Marius Muja and David G Lowe. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration." In: ().

[104] Raul Mur Artal and Juan D. Tardos. “Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM”. In: *Proc. Robotics: Science and Systems (RSS)*. Rome, Italy, 2015.

[105] Venkatraman Narayanan and Maxim Likhachev. “Discriminatively-guided Deliberative Perception for Pose Estimation of Multiple 3D Object Instances”. In: *Proceedings of Robotics: Science and Systems*. AnnArbor, Michigan, 2016. DOI: `10.15607/RSS.2016.XII.023`.

[106] Venkatraman Narayanan and Maxim Likhachev. “PERCH: Perception via Search for Multi-Object Recognition and Localization”. In: *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE. 2016.

[107] E.R. Nascimento et al. “BRAND: A robust appearance and depth descriptor for RGB-D images”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS' 2012)*. 2012, pp. 1720–1726.

[108] John A Nelder and Roger Mead. “A simplex method for function minimization”. In: *The computer journal* 7.4 (1965), pp. 308–313.

[109] Shashank Pathak et al. “Robust Active Perception via Data-association aware Belief Space Planning”. In: *arXiv preprint arXiv:1606.05124* (2016).

[110] Sachin Patil et al. “Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation”. In: *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 515–533.

[111] Timothy Patten et al. “Viewpoint evaluation for online 3-d active object classification”. In: *IEEE Robotics and Automation Letters* 1.1 (2016), pp. 73–81.

[112] James Philbin et al. “Object retrieval with large vocabularies and fast spatial matching”. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8.

[113] Richard Pito. “A sensor-based solution to the “next best view” problem”. In: *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. Vol. 1. IEEE. 1996, pp. 941–945.

[114] Richard Pito. “A solution to the next best view problem for automated surface acquisition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.10 (1999), pp. 1016–1030.

[115] R. Platt et al. "Belief space planning assuming maximum likelihood observations". In: *Proceedings of Robotics: Science and Systems*. Zaragoza, Spain, 2010. DOI: 10.15607/RSS.2010.VI.037.

[116] Robert Platt Jr et al. "Belief Space Planning Assuming Maximum Likelihood Observations". In: *Proceedings of the Robotics: Science and Systems Conference, 6th*. 2010.

[117] Christian Potthast and Gaurav S Sukhatme. "A probabilistic framework for next best view estimation in a cluttered environment". In: *Journal of Visual Communication and Image Representation* 25.1 (2014), pp. 148–164.

[118] Christian Potthast et al. "Active multi-view object recognition: A unifying view on online feature selection and view planning". In: *Robotics and Autonomous Systems* 84 (2016), pp. 31 –47.

[119] Samuel Prentice and Nicholas Roy. "The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance". In: *International Journal of Robotics Research* 28.11-12 (2009), pp. 1448–1465.

[120] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *arXiv preprint arXiv:1506.02640* (2015).

[121] Shaoqing Ren et al. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[122] Andreas Richtsfeld et al. "Segmentation of unknown objects in indoor environments". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 4791–4796.

[123] Edward Rosten and Tom Drummond. "Machine learning for high-speed corner detection". In: *Proc. European Conference on Computer Vision (ECCV'2006)*. Springer, 2006, pp. 430–443.

[124] Nicholas Roy and Samuel Prentice. "The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance". In: *The International Journal of Robotics Research* (2009). DOI: 10.1177/0278364909341659.

[125] Sumantra Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. "Active recognition through next view planning: a survey". In: *Pattern Recognition* 37.3 (2004), pp. 429–446.

[126] Ethan Rublee et al. “ORB: an efficient alternative to SIFT or SURF”. In: *Proc. IEEE International Conference on Computer Vision (ICCV'2011)*. IEEE. 2011, pp. 2564–2571.

[127] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. “Fast point feature histograms (FPFH) for 3D registration”. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 3212–3217.

[128] Radu Bogdan Rusu and Steve Cousins. “3d is here: Point cloud library (pcl)”. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1–4.

[129] Radu Bogdan Rusu et al. “Persistent point feature histograms for 3d point clouds”. In: *Proc. International Conference on Intelligent Autonomous Systems (IAS'2008)*. 2008.

[130] R.B. Rusu et al. “Fast 3D recognition and pose using the Viewpoint Feature Histogram”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2010)*. 2010, pp. 2155–2162.

[131] John Schulman et al. “Motion planning with sequential convex optimization and convex collision checking”. In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1251–1270.

[132] P. Sermanet, K. Kavukcuoglu, and Y. LeCun. “EBLearn: Open-Source Energy-Based Learning in C++”. In: *2009 21st IEEE International Conference on Tools with Artificial Intelligence*. 2009, pp. 693–697. DOI: 10.1109/ICTAI.2009.28.

[133] Arjun Singh et al. “Bigbird: A large-scale 3d database of object instances”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 509–516.

[134] Josef Sivic and Andrew Zisserman. “Video Google: A text retrieval approach to object matching in videos”. In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE. 2003, pp. 1470–1477.

[135] N Snavely, S Seitz, and R Szeliski. “Photo Tourism: Exploring Image Collections in 3D. ACM Transactions on Graphics”. In: *ACM Transactions on Graphics* (2006).

[136] Carsten Steger. “Similarity measures for occlusion, clutter, and illumination invariant object recognition”. In: *Joint Pattern Recognition Symposium*. Springer. 2001, pp. 148–154.

[137] Michael Steinbach, Levent Ertöz, and Vipin Kumar. “The challenges of clustering high dimensional data”. In: *New directions in statistical physics*. Springer, 2004, pp. 273–309.

[138] Jie Tang et al. “A textured object recognition pipeline for color and depth image data”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 3467–3474.

[139] Timothy B Terriberry, Lindley M French, and John Helmsen. “GPU accelerating speeded-up robust features”. In: Citeseer.

[140] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[141] Emanuel Todorov and Weiwei Li. “A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems”. In: *Proceedings of the 2005, American Control Conference, 2005.* IEEE. 2005, pp. 300–306.

[142] F. Tombari, S. Salti, and L. Di Stefano. “A combined texture-shape descriptor for enhanced 3D feature matching”. In: *Proc. IEEE International Conference on Image Processing (ICIP’2011)*. 2011, pp. 809–812.

[143] Federico Tombari, Samuele Salti, and Luigi Di Stefano. “Performance evaluation of 3D keypoint detectors”. In: *International Journal of Computer Vision* 102.1-3 (2013), pp. 198–220.

[144] Federico Tombari, Samuele Salti, and Luigi Di Stefano. “Unique Signatures of Histograms for Local Surface Description”. In: *Proc. European Conference on Computer Vision (ECCV’2010)*. 2010, pp. 356–369.

[145] Panu Turcot and David G Lowe. “Better matching with fewer features: The selection of useful features in large database recognition problems”. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 2109–2116.

[146] Matthew Turk and Alex Pentland. “Eigenfaces for recognition”. In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86.

[147] Ranjith Unnikrishnan and Martial Hebert. “Multi-scale interest regions from unorganized point clouds”. In: *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW’08. IEEE Computer Society Conference on*. IEEE. 2008, pp. 1–8.

[148] Jur Van Den Berg, Pieter Abbeel, and Ken Goldberg. “LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information”. In: *The International Journal of Robotics Research* 30.7 (2011), pp. 895–913.

[149] Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. "Motion planning under uncertainty using iterative local optimization in belief space". In: *The International Journal of Robotics Research* 31.11 (2012), pp. 1263–1278.

[150] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[151] Hough Paul VC. *Method and means for recognizing complex patterns*. US Patent 3,069,654. 1962.

[152] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. `http://www.vlfeat.org/`. 2008.

[153] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2001, pp. I–511.

[154] Michael W Walker, Lejun Shao, and Richard A Volz. "Estimating 3-D location parameters using dual number quaternions". In: *CVGIP: image understanding* 54.3 (1991), pp. 358–367.

[155] Jinjun Wang et al. "Locality-constrained linear coding for image classification". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 3360–3367.

[156] Stefan Wenhardt et al. "An information theoretic approach for next best view planning in 3-d reconstruction". In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 1. IEEE. 2006, pp. 103–106.

[157] Peter Whaite and Frank P Ferrie. "Autonomous exploration: Driven by uncertainty". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.3 (1997), pp. 193–205.

[158] Changchang Wu.

[159] Changchang Wu. *SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT)*. `http://cs.unc.edu/~ccwu/siftgpu`. 2007.

[160] Kanzhi Wu, Ravindra Ranasinghe, and Gamini Dissanayake. "Active recognition and pose estimation of household objects in clutter". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 4230–4237.

[161] Jianchao Yang et al. “Linear spatial pyramid matching using sparse coding for image classification”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 1794–1801.

[162] Jun Yang et al. “Evaluating bag-of-visual-words representations in scene classification”. In: *Proceedings of the international workshop on Workshop on multimedia information retrieval*. ACM. 2007, pp. 197–206.

[163] Yu Zhong. “Intrinsic shape signatures: A shape descriptor for 3d object recognition”. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 689–696.