Using 3D Visual Data to Build a Semantic Map for Autonomous Localization

by

Qiang Liu

A thesis submitted for the degree of

Doctor of Philosophy

School of Computer Science and Electronic Engineering

University of Essex

Date of submission: July 2018

Abstract

Environment maps are essential for robots and intelligent gadgets to autonomously carry out tasks. Traditional maps built by visual sensors include metric ones and topological ones. These maps are navigation-oriented and not adequate for service robots or intelligent gadgets to interact with or serve human users who normally rely on conceptual knowledge or semantic contents of the environment. Therefore, semantic maps become necessary for building an effective human-robot interface. Although researchers from both robotics and computer vision domains have designed some promising systems, mapping with high accuracy and how to use semantic information for localization remain challenging.

This thesis describes several novel methodologies to address these problems. RGB-D visual data is used as system input. Deep learning techniques and SLAM algorithms are combined in order to achieve better system performance. Firstly, a traditional feature based semantic mapping approach is presented. A novel matching error rejection algorithm is proposed to increase both loop closure detection and semantic information extraction accuracy. Evaluational experiments on public benchmark dataset are carried out to analyze the system performance. Secondly, a visual odometry system based on a Recurrent Convolutional Neural Network is presented for more accurate and robust camera motion estimation. The proposed network deploys an unsupervised end-to-end framework. The output transformation matrices are on an absolute scale, i.e. true scale in the real world. No data labeling or matrix post-processing tasks are required. Experiments show the proposed system outperforms other state-of-the-art VO systems. Finally, a novel topological localization approach based on the pre-built semantic maps is presented. Two streams of Convolutional Neural Networks are applied to infer locations. The additional semantic information in the maps is inversely used to further verify localization results. Experiments show the system is robust to viewpoint, lighting condition and object changes.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Huosheng Hu. He is a man full of wisdom. He is always passionate and very kind to everyone around him. Many thanks for his academic advices on my research. It is him who guided me into the robotics field and showed me how to start a research, from choosing a topic until writing this thesis. Many thanks for his patience in countless discussions on the research methodologies. Apart from his guidance on my research, I would also like to thank him for helping me develop other generic skills that will be useful in my future career, e.g., language competence, presentation and demonstration skills, teaching skills and how to carry out a project. Many thanks for his financial support during the past four year. His passion about both research and life will inspire me in my whole life.

Beside my supervisor, I would like to thank Prof. Dongbing Gu, who also gives me a lot of advice on my research and study. He is always calm and offers his honest suggestions on my daily life. Many thanks to Prof. Huijun Gao, who supports me to continue my PhD study in the UK. Many thanks to my board meeting chief, Dr. Sam Steel, for checking my research progress and giving me useful advices. My thanks also go to Mr. Robin Dowling and Mr. Ian Dukes for their kind assistant with various technical problems when I carried out my experiments, as well as countless tea times in the afternoon. My sincere thanks also goes to Dr. Sen Wang and Dr. Ling Chen, who give me useful academic advices and help me settle down in the UK. Many thanks to my close friend, Dr. Ruihao Li, who provides great help for each of my publications. His lifestyle also inspires me to get rid of my bad habits.

I thank my fellow lab-mates, my friends in the UK for their accompany during my PhD study. Thank Fang Wang, Yan Zhuang, Yisha Liu, Yi An, Yiming Xu, Guan Lu, Guochen Wang, Wenju Zhou, Hong Liu, Jing Gao, Xunyu Zhong, Juan Yu, Yuanping Xie, Lisheng Wei, Chao Zhang, Jianjun Gui, Cheng Zhao, Shuofei Yang, Wanxin Zhang, Qi Zhang, Can Zuo, Tong Pei, Xiaoran Yang, Qian Wang, Kezhi Wang, Tianlang Peng, Yaser Ibrahem, Wesam Jasim, Minhuan Guo, Jialin Liu, Aakash Soni, Dingtian Yan, Zuyuan Zhu, Tuo Feng, Xiaoqi Nong, Penny Roberts, Xiaozhen Cai, Mark Boo, Alexander Jegorov, Xinye Zhang, Kikillo Secas, Gonzillah Monstah, Dunja Mrvic, Mihail Ciobanu, etc., for bringing joy to my life and encouraging me to moving forward each day.

Last but not the least, I would like to thank my family. Many thanks to my parents for raising me up and supporting me spiritually throughout writing this thesis and my life in general.

Contents

List of Figures

List of Tables

List of Algorithms

Notation and Nomenclature

s	scale factor
p_i, q_i	a pair of matched points
Q_i	associated 3D point of q_i
F_p	RGB image with good matches
F_Q	depth image with good matches
(u_{qi}, v_{qi})	2D coordinates of a point in a monocular image
(x_{Qi}, y_{Qi}, z_{Qi})	3D coordinates of a point in a point cloud
(f_x, f_y)	focal lengths expressed in pixel units
(c_x, c_y)	a principal point that is usually at the image center
$\hat{R}$	estimated rotation matrix
$\hat{T}$	estimated translation matrix
k_i	camera pose
e	error function
N_c	number of good matches in the candidate key-frame
N_n	number of good matches in the neighbouring key-frame
ξ_m	ratio of good matches in the candidate and neighbouring key-frame
I_t	monocular image
D_t	depth image
d_t	depth value of a point in the depth image
p_t	a pixel in a monocular image
$\hat{p}_t$	projected point
P_t	a voxel in a point cloud
K	camera intrinsic matrix

L_{2D}	2D spatial loss
C_t	point cloud
L_{3D}	3D spatial loss
L	total loss
λ_{2D}	weight for 2D spatial loss
λ_{3D}	weight for 3D spatial loss
γ	Gamma Correction
s_x	scale factor along X-axis
s_y	scale factor along Y-axis
r_d	rotation in degrees
β_1	exponential decay rate for the first moment estimate
β_2	exponential decay rate for the second moment estimate
N	sequence length
l_i	location
x	a query image
$P(l_i)$	place recognition predicted score
$P(x)$	probability of the object existing in the image
$P(x\|l_i)$	empirical knowledge
$P(l_i\|x)$	rectified score
$P(x\|\boldsymbol{L})$	empirical probability distribution
$P(\boldsymbol{L}\|x)$	normalized distribution
y	labeled object
ξ	a ratio that controls the weights of two CNN stream outputs

List of Acronyms

SYSIASS	Autonomous and Intelligent Healthcare System
COALAS	Cognitive Assisted Living Ambient System
RGB-D	Red, Green, Blue and Depth
ToF	Time of Flight
HOG	Histogram of Oriented Gradient descriptor
DOT	Dominant Orientation Templates
GIST	a global image descriptor which summarizes the “gist” of a scene
LINE-MOD	Multimodal Line
KLT	Kanade Lucas Tomasi
FAST	Features from Accelerated Segment Test
FAST-ER	Faster and Better corner detector
AGAST	Adaptive and Generic Accelerated Segment Test
BRIEF	Binary Robust Independent Elementary Features
LoG	Laplacian of Gaussian
PDE	Partial Differential Equations
DoG	Difference of Gaussians
DoH	Determinant of the Hessian
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
GLOH	Gradient Location and Orientation Histogram
SURF	Speeded Up Robust Features
CenSurE	Centre Surround Extrema
ORB	Oriented FAST and Rotated BRIEF

BRISK	Binary Robust Invariant Scalable Keypoints
FREAK	Fast Retina Keypoint
MSER	Maximally Stable Extremal Region
HMM	Hidden Markov Model
AdaBoost	Adaptive Boosting
SVM	Support Vector Networks
SIMD	Single Instruction, Multiple Data
SSE	Streaming SIMD Extensions
KD-tree	K-Dimensional tree
RFs	Random decision Forests
BoW	a Bag of Binary Words
OCR	Optical Character Recognition
KR	Knowledge Representation
AI	Artificial Intelligence
NBC	Naive Bayesian Classifier
OWL-DL	Web Ontology Language-Description Logic
CNN	Convolutional Neural Network
DoF	Degree of Freedom
RCNN	Recurrent Convolutional Neural Network
PR2	a robotics research platform from Personal Robotics
AR	Augmented Reality
g2o	a general framework for Graph Optimization
RANSAC	Random Sample Consensus
FLANN	Fast Library for Approximate Nearest Neighbor
PCL	Point Cloud Library
CPU	Central Processing Unit
GPU	Graphics Processing Unit
ATE	Absolute Trajectory Error
RMSE	Root-Mean-Square Error
VO	Visual Odometry
IMU	Inertial Measurement Unit

LiDAR	Light Detection and Ranging
KITTI	a vision benchmark suite
PTAM	Parallel Tracking and Mapping
DTAM	Dense Tracking and Mapping
LSD-SLAM	Large-Scale Direct Monocular SLAM
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
VGG	Visual Geometry Group neural network
ReLU	Rectified Linear Unit
ELU	Exponential Linear Unit
ICP	Iterative Closest Point
SfM	Structure from Motion
ImageNet	an image database
ILSVRC	ImageNet Large Scale Visual Recognition Challenge

Chapter 1

Introduction

This research is focused on building a semantic map through low-cost cameras and using the map for localization. Both feature based and deep learning based methods are studied. Novel approaches are proposed. This chapter describes the research motivations, objectives, challenges, research methodologies, thesis contributions and thesis outline.

1.1 Motivations

In recent years, the steady increase in life expectancy and lower birth rate, all together, have caused a significant raise in the percentage of the elderly and disabled people in the world. This has, in turn, led to a major challenge for both developed and developing countries to enable them to remain at a state of independence. Therefore, numerous projects, which aim to improve the quality of life for people suffering from a loss of autonomy through age, illness or accidents, are carried out globally, such as Sweet-Home [6], SYSIASS [7], COALAS [8], etc. These projects involve interaction of various electromechanical components and systems which are coordinated. To some extent, these projects have potentially promoted the development in service robots, some of which are even designed to become a part of the life of ordinary people [9] and have the ability to communicate with us.

On the other hand, wearable technology has expended rapidly [10]. At first, researchers focused on solutions proposed for the detection of risk situations and the automatic analysis of behavioral disorders. Most of these solutions are based on embedded technology, in which case electronic gadgets are embedded into the belts and clothes of users, e.g., Tadiran [11] and Vigilio [12]. In this approach, the gadgets may contain a combination of different measuring sensors, such as temperature, heart rate, accelerometers and gyroscopes. These devices facilitate pertinent analysis of a fragile state of a person. More recently, research on wearable products becomes a new trend, some of which are even able to communicate with people. For example, the electronic travel aids [13] can provide the blind with the ability of localization and navigation as well as semantic guidance.

It becomes very clear that robotics is working its way into our lives in an attempt to fulfill our needs for household servants, health tracking devices and even cognitive companions. Apart from accurate navigation and manipulation performance, robots are also required to understand, interpret and represent dynamic environments autonomously. Furthermore, they also have to interact with people in a human-compatible way. Thus the acquisition of spatial models of physical environments, as well as the interpretation from sensor data to semantic information are the prerequisites to the pursuit of building truly autonomous robots and assistive devices.

The human-robot interface has always been a challenging topic in the field of robotics. A truly intelligent robot should have the capability of perceiving, understanding, interpreting and representing the dynamic and complex environments in order to communicate with us. Generally, semantic mapping is regarded as an efficient solution to address this problem. A semantic map plays a key role and meets the demand by representing not only spatial dimensions but also conceptual knowledge of the environment. A direct method to obtain rich information is through visual data since humans perceive the world through our eyes.

In order to build a semantic map of a scene with visual data, camera poses need to be estimated and sequentially connected first. The objects or places in an environment are then detected and recognized, which serve as semantic information in a semantic map. Traditional solutions for both processes rely on image features. Global or local image features are designed or selected, detected and described by feature descriptors. However, traditional visual features are hand-crafted and thus quite limited to appearance such as edges, corners, which are difficult to be generalized. In addition, such systems have to be carefully designed and fine-tuned to achieve optimal performance. This normally requires huge engineering effort. These limitations suggest us to seek other less labour intensive solutions.

The deep learning technique developed in recent years has shown some promising results. Inspired by information processing and communication patterns in biological nervous systems, a deep learning based network can learn from large amount of data by creating a more abstract, general and robust representation as the network grows deeper. As a result, the model automatically extracts features and yields higher accuracy results, thus making deep learning dominate many tasks, e.g., object recognition and detection, natural language processing, mobile advertising, etc. In this thesis, deep learning is leveraged to address some challenging issues.

In conclusion, the motivation of my research is to build a semantic map for mobile robots or people suffering from vision loss based on visual data in order to help them move around or carry out tasks more easily. The additional semantic information in a map can then be further utilized to boost localization accuracy.

1.2 Objectives and Challenges

Motivated by the importance of semantic mapping and localization, this thesis aims at using 3D visual data to build a semantic map and then using the semantic map for topological localization. Both theoretical and experimental analyses are provided on the proposed methodologies.

To achieve this goal, this thesis will focus on the following objectives.

- **High accuracy:** Scientific research should be conducted accurately. Accuracy measures how close the performance of a proposed method comes to its true value. It is one of the most important criteria to evaluate the quality of a system. Therefore, one objective of this thesis is to achieve high accuracy for both semantic mapping and localization methods [14].
- **Strong robustness:** When a robotic system is put into practice, the raw input can be noisy. Therefore, the system should have strong robustness, i.e., the capacity to cope with erroneous input and remain unaffected during normal usage [15]. Robustness measures the reliability of a system, which is another objective of this thesis.
- **Good autonomy:** Before a system can be applied, its structure needs to be carefully designed. The parameters need to be auto-tuned, especially when a system is used in a unfamiliar new environment. Even in a deep learning based system, manually labeling training data is sometimes inevitable. Engineering effort evaluates how easy a system can be used and how autonomous a system is, which is also one of our objectives to minimize [16].
- **Low cost and small in size:** Robotic systems need to be low cost and small in size if they

can be widely used in our daily life [17]. It is extremely important to create such sensors and robotic devices for general public to use, which is one of our research objectives.

In order to achieve the aforementioned objectives, various challenges emerge and have been addressed in this thesis.

(1) **How to generate semantic information with high accuracy.**
Traditional object or place recognition methodologies are based on the features detected and described by feature descriptors. During semantic mapping, local visual features which are robust to geometric transformations and illumination changes are widely applied. However, the features are quite limited to the appearance of objects such as edges, corners and their relations [18]. Matching errors inevitably exist. Moreover, a robot which can answer the question "Is this my kitchen?" has been addressed by several works using diverse sensor data. However, a truly autonomous robot should be able to answer a harder and more general question "Is this a kitchen?" The problem then comes down to the design of a more general classifier which can classify objects and places by conceptualization. This might be the first definite step to enable a robot to behave in a manner that is compatible with humans [19].

(2) **How to improve the camera pose estimation performance.**
Generally speaking, the performance of the mapping process is highly dependent on the accuracy of the camera pose estimation algorithm [20] as a global environment map is built by connecting sequential images. Traditional pose estimation systems adopt a classic pipeline, i.e., feature detection and description, feature matching, transformation estimation, loop closure detection, global pose graph optimization [21]. Researchers have proposed various algorithms for this pipeline. Each step has significant influence on the overall system performance. Such systems are usually computationally efficient since only the features of an image are involved in calculation. However, designing a traditional feature based system requires intensive labour and some hidden features of an image are abandoned from the beginning [22]. Thus, we need to seek other end-to-end solutions for camera pose estimation.

(3) **How to use additional semantic knowledge to enhance localization capability.**
Semantic knowledge allows us to explicitly account for perceptual aliasing when merging

the metric data into the global spatial layer. Map building is a continuous long-time process during which precise metric data is obtained from visual sensors. Similar scenes inevitably exist, thus localization errors appear if similar image features are detected when we revisit a scene [23]. This may result in a significant jump in the topological map, i.e., localizing the robot in another irrelevant location with highly similar appearances to the current location. Apart from the geometrical optimization methods that have already been published, the additional semantic knowledge in a semantic map should also be utilized to infer locations based on the relations between detected objects and places zcite[24, 25, 26]. Once an object is detected and inferred irrelevant to the current place on the basis of common sense knowledge, the robot should be able to reason about potential localization errors.

1.3 Research Methodologies

In order to conquer the challenging issues outlined in the previous section, a number of novel research methodologies are proposed in this thesis.

To extract semantic information with higher accuracy, we try to improve the performance of traditional feature based semantic mapping systems. Image features are extracted beforehand for camera pose estimation and object recognition. However, matching accuracy during object recognition is quite low, even when the experiments are carried out in a small environment. Therefore, a novel matching error elimination algorithm is introduced for both loop closure detection and object recognition, which increases the performance of traditional semantic mapping approach. Moreover, deep learning is also leveraged for object recognition, which improves object recognition accuracy during semantic information extraction. On the other hand, it expands the range of identifiable items in an environment. The size of the database is significantly minimized since only object names are stored in the database rather than object images.

To address the second challenge, a novel monocular visual odometry system using an unsupervised end-to-end Recurrent Convolutional Neural Network is proposed. Fine-tuning hyper parameters and data labeling tasks, which can be quite time-consuming and labour intensive are not required. The RCNN consists of a CNN and a LSTM network. 2D and 3D spacial losses are designed based on warping and inverse warping technique. We evaluate the system performance on KITTI Odometry dataset. Both qualitative and quantitative analyses are carried out and results show that the proposed system not only saves engineering effort, but also improves the camera pose estimation accuracy.

To tackle the third challenge and boost the topological localization performance, a novel system consisting of two streams of CNNs is proposed. The two CNNs are separately trained. One is used for place recognition and the other one is used for object detection. In this way, when an accurate semantic map is obtained, the semantic information can be employed to further verify the localization result by detecting distinctive objects within the input image. Experiments are carried out in two indoor environments and the localization performance in terms of appearance variations such as viewpoint, lighting condition and object changes are analyzed in detail. Results show that both the precision and recall rates are improved.

1.4 Thesis Contributions

The major contributions of this thesis are briefly listed as follows.

(1) To create a feature based semantic mapping system. The geometrical mapping pipeline consists of SURF feature extraction, feature matching, camera pose estimation, loop closure detection and global pose graph optimization. Semantic information is extracted by matching a new key-frame to the object images in the database.

(2) A novel matching error elimination method is proposed for loop closure detection and semantic information extraction. The ratio of good matching numbers in the current key-frame to those in the neighbouring key-frames is used for outlier rejection. Experiments have shown the proposed method can improve the semantic mapping performance.

(3) A novel deep learning based visual odometry system is developed, which is composed of a CNN and a RNN. The training strategy is based on an unsupervised end-to-end manner. Thus, no labeling task is needed and no ground truth camera poses are required for training. Experiments have shown the proposed system outperforms other state-of-the-art VO systems.

(4) Absolute scale recovery is achieved from only monocular images. Since both monocular images and depth information are used for training, absolute scale is thus injected into the RCNN. Therefore, no pose post-processing is required. 2D and 3D spacial losses are

both deployed to punish the output deviation from the truth. In this way, we maximize the benefit of the input RGB-D data.

(5) A semantic information extraction method is developed based on deep learning technique. Hence, the recognition accuracy is increased, the range of identifiable items is broadened and the size of the semantic information database is reduced.

(6) A novel topological localization approach is created based on pre-built semantic maps. Two CNN steams are used and semantic information in the maps are inversely deployed to further verify the localization results. Experiments have shown the proposed approach improves the localization performance in terms of both precision and recall.

1.5 Thesis Outline

A high level overview of the structure of this thesis is presented in Figure 1.1. This chapter gives the research motivations, objectives, challenges addressed, research methodologies. Chapter 2 form the base of this work. The main work and contributions are presented in Chapter 3, 4 and 5. Chapter 6 concludes the thesis.

Chapter 2 reviews recent research and development of semantic mapping and localization based on visual sensors. Both traditional feature based approaches and recent deep learning based approaches are presented. Semantic information extraction approaches are discussed. Semantic representation methods are subsequently outlined. Finally, some real-world indoor and outdoor applications are given.

Chapter 3 describes a traditional feature based semantic mapping approach. A classic mapping pipeline is introduced. A novel matching error elimination algorithm is then introduced for both loop closure detection and object recognition. Finally, we carry out some experiments in a student accommodation and compare the proposed method to other state-of-the-art algorithms on the public TUM RGB-D SLAM dataset.

Chapter 4 proposes a novel monocular visual odometry system based on an unsupervised end-to-end RCNN framework. The network is introduced and the loss functions are detailed. Training strategy is then presented including image augmentation methods, hyper parameters and some training tricks. Finally, we evaluate the system performance on KITTI Odometry dataset.

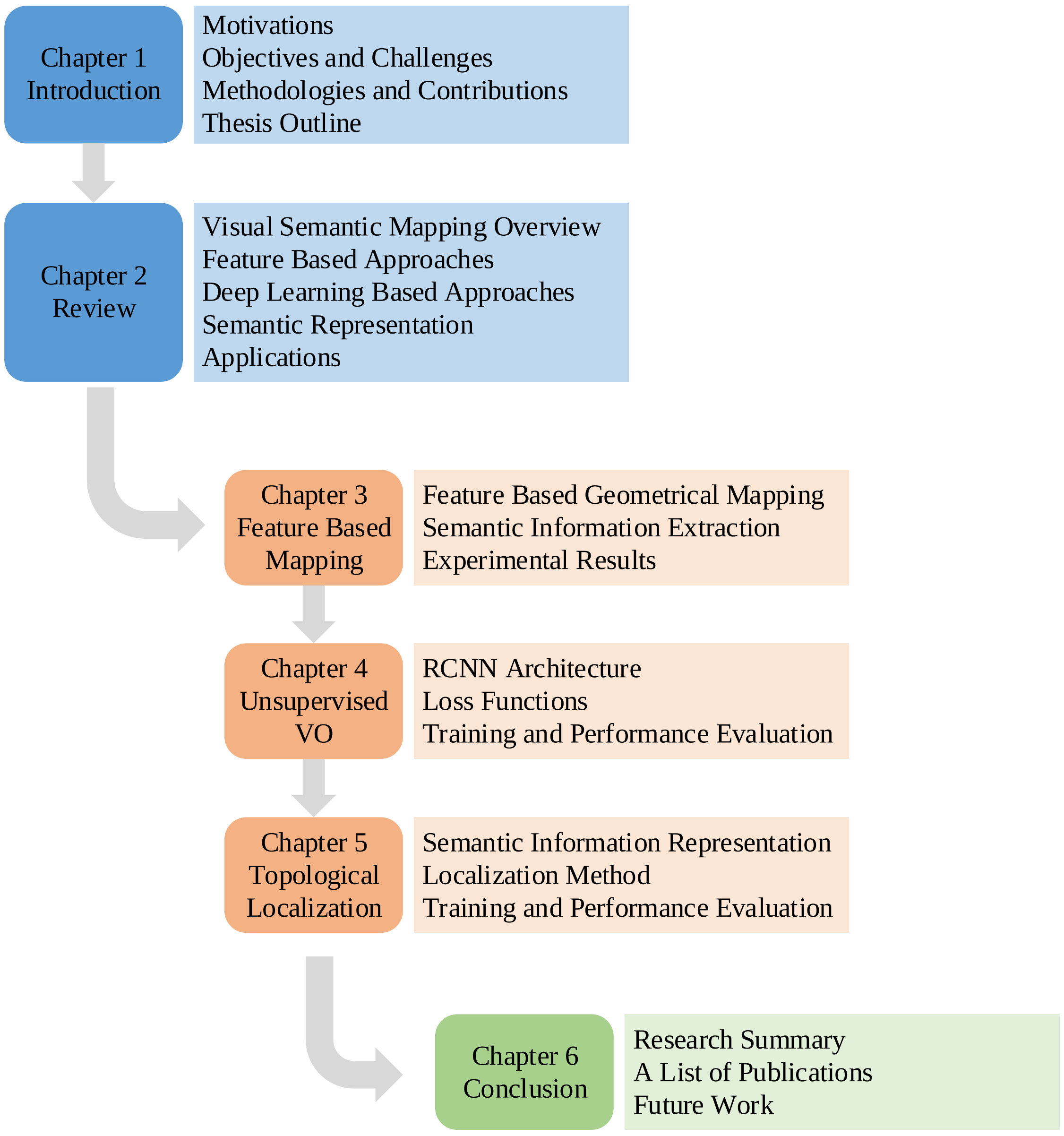


Figure 1.1: Structure of this thesis.

Chapter 5 introduces a novel topological localization method based on deep learning technique. Semantic information extraction and representation approaches are first presented. The topological localization method based on two separately trained CNNs is then presented. Experiments are carried out in two indoor environments. Results of the proposed method are compared to those of other state-of-the-art algorithms.

Chapter 6 summarizes the presented work and major contributions. The conference and journal papers published or submitted during this research are then listed. Finally, potential works that can be carried out in the future are given.

Chapter 2

Background Review

2.1 Introduction

Traditionally, robotic mapping is broadly divided into two categories - metric and topological mapping. Metric maps describe the geometric features of an environment, whereas topological maps involve the connectivity of different places and are used for robots to navigate from one place to another [27]. Figure 2.1 and Figure 2.2 show a metric map and a topological map respectively. An early representative of the metric mapping approach is based on occupancy grids that model the occupied and free space. In contrast, the topological mapping approach uses nodes to represent distinct places or landmarks, and curve lines to describe the paths between nodes. Recently, a new hybrid mapping method that combines metric and topological paradigms is developed to compensate for the weaknesses of individual approaches. This mapping approach applies a metric map for accurate navigation in a local space, and a global topological map for moving from one place to another.

All these traditional mapping approaches are navigation-oriented and enable mobile robots to navigate around and plan a path to reach a goal [28]. The maps built by traditional mapping approaches are relatively low-level since they are unable to interpret scenes or encode semantic information. To serve people, service robots should be able to communicate with humans through semantic information such as human speech commands, “Can I have a cup of coffee?” or “Please open the window”, so that they are able to interact with humans in a human-compatible way [29].

In a semantic map, nodes representing places and landmarks are named by linguistic words.

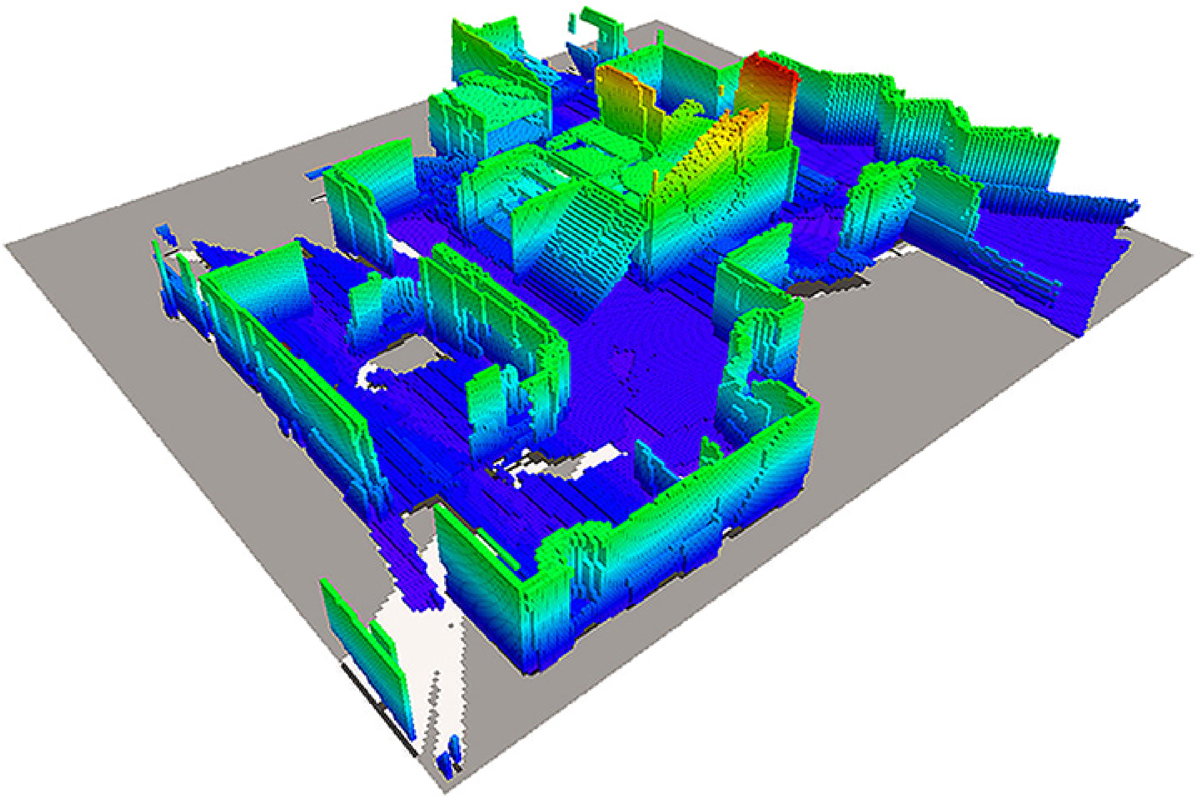


Figure 2.1: Metric map [1].

Examples of these include names and categories of different objects, rooms and locations. More specifically, a semantic map can be regarded as an extension of a hybrid map, which contains geometric description, topological connectivity and semantic interpretation [30]. It provides a friendly way for robots to communicate with humans. This section reviews numerous publications in visual based semantic mapping and attempts to provide an overview of the state-of-the-art methodologies in this field. It is mainly focused on how to extract semantic information from visual data, including feature extraction, object/place recognition and semantic representation and deep learning based methods. It differs from other existing comprehensive surveys on traditional robot mapping approaches [27] or general semantic mapping methodologies [31].

Recently, using semantic data to represent environments has become a popular research domain and drawn enormous attentions from different fields [32, 33]. The application of visual data in semantic mapping systems seems to be a sensible decision as humans perceive the world through eyes. Visual data allows the representation of both low-level features such as lines, corners and shapes, and high-level features such as colours, relations and texts. In this way, a

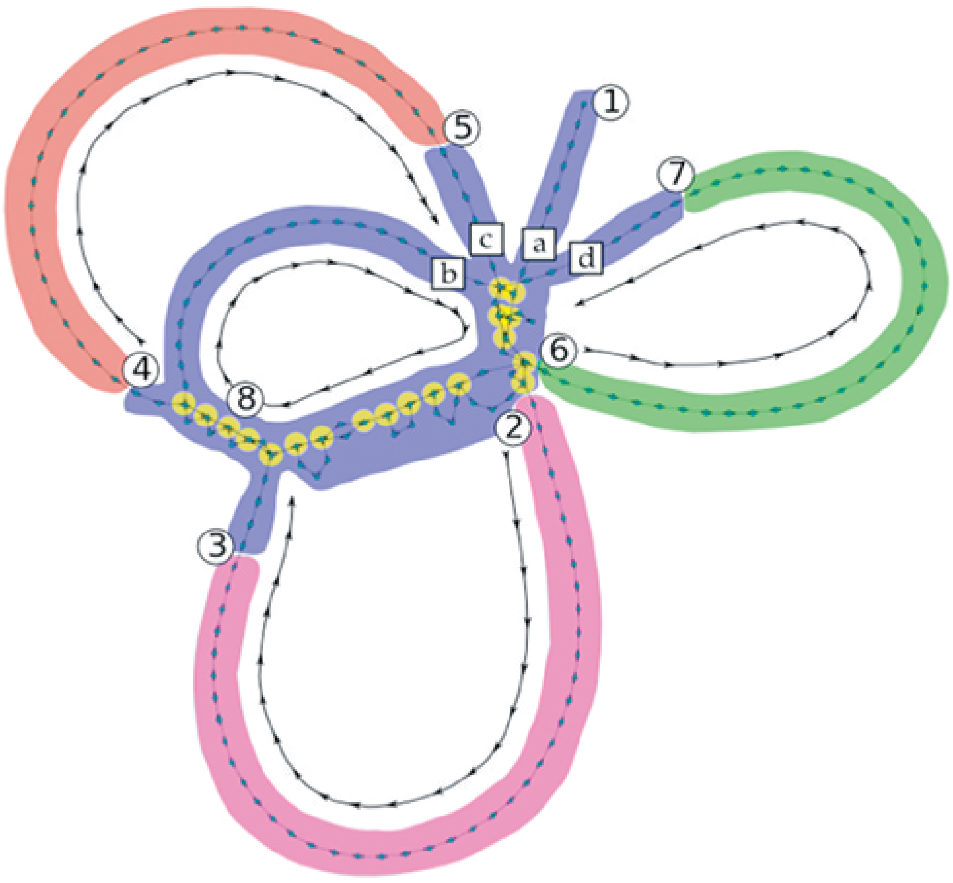


Figure 2.2: Topological map [2].

wider variety of objects can be recognized, which can highly enrich semantic maps.

In general, a traditional visual based semantic mapping system consists of three parts. At first, the specific features are pre-selected based on sensor type, and feature descriptors are computed and obtained. Subsequently, features or descriptors are classified in terms of prior knowledge so that objects and places can be recognized. Finally, properties are endowed with semantic meanings on the basis of topological and metric maps. Figure 2.3 presents the general process for semantic mapping. Note that a metric map is considered as a complementary attribute of a semantic map. In addition, some systems rely on direct image segmentation to obtain semantic information rather than using feature descriptors to represent objects or scenes.

Many types of visual sensors have been developed to provide a variety of interpretations of the world. In addition, the subsequent processing methods are highly dependent on the data type used. To some extent, the visual sensor applied plays a key role in a semantic mapping system. The visual sensors used for robot mapping include conventional monocular cameras,

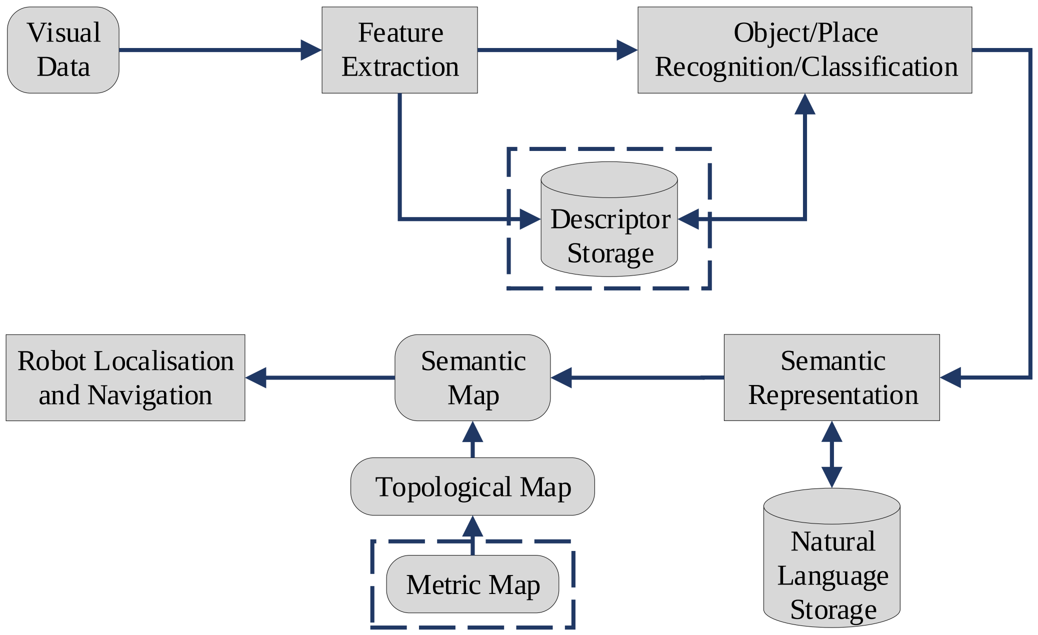


Figure 2.3: Overview of the general process for semantic mapping.

omni-directional cameras, stereo cameras and RGB-D cameras. At first, visually recognizing objects or places was normally done by using conventional cameras that record two dimensional images. In recent years, extracting semantic information from 3D point clouds has become a new trend due to the availability of low-cost and light-weighted 3D point cloud capturing devices such as stereo cameras and RGB-D sensors, which allow the application to small robot platforms or even wearable devices easily. Compared with 2D images, 3D point clouds overcome the limitation in the data-stream itself by providing additional depth data. Moreover, humans recognize and perceive a 3D world in terms of our eyes. Therefore, object recognition through capturing 2D projections of the 3D world is inevitably inaccurate and might be even misleadingly suggested, especially when it comes to a large variety of goods in our daily life [34].

The rest of this section is organized as follows. In Section 2.2, visual feature extraction methodologies are outlined and classified in terms of global and local features. Section 2.3 describes three basic recognition approaches in semantic mapping, namely global, local and hybrid approaches. More direct deep learning based approaches developed in recent years are given in this section. Subsequently, how to generate semantic representations of the environment is outlined in Section 2.4 and some typical real-world applications are presented in Section

2.5. Finally, a brief conclusion and discussion are given in Section 2.6.

2.2 Visual Features Used in Semantic Mapping Systems

In the last decade, some researchers have reported systems in which semantic interpretation of certain scenes were obtained [35, 36]. However, the acquisition was done through conversations between humans and robots or even hand-coded into the systems rather than using the robots' own sensors [37]. Visual features describe the elementary characteristics such as shapes, colours, textures, motions and relations between pixels in raw visual data and can be broadly divided into two categories: global and local features [38].

Global features represent an image as a whole without directly describing the spatial layout of the properties in the image. More specifically, the statistics of all the pixels in a movable fixed-size bounding box are extracted to generate feature vectors which can determine the likelihood for image matches. Such features are suitable for large scale environment recognition, e.g. roads, lawns in outdoor environments and rooms in buildings. However, global features are sensitive to cluttered background and occlusion due to their essential attributes. Therefore, their performance drops relatively in the case of object recognition in indoor environments where direct specification of the content in an image is required or when an object is not enclosed by the bounding box. Local features on the other hand rely on individual pixels or discrete regions. Typically, salient features of highly textured objects are extracted by feature detectors and represented by compendious feature descriptors. The representation of the content in an image is thus more robust to scale, viewpoint or illumination changes.

2.2.1 Global Features

Despite the limitation of global features, they are still useful in cases where a rough place or object classification is required. Global features consist of the statistics extracted from a whole image or a bounding box, such as contour, shape, texture, colour or a combination of them [39]. They generally represent an image with a single high-dimensional feature vector, and thus can be easily applied with any standard classification methods [40]. Moreover, thanks to the compact representation and low computational cost, they have been employed by some semantic mapping systems in real-time. Table 2.1 presents the differences and similarities of the global features used for object recognition in semantic mapping systems.

Table 2.1: Global features for object recognition in semantic mapping systems.

		Feature	**Performance**
Detector	Texture	Haar-like [41]	Robust to illumination changes
Descriptor	Color	Color histograms [42]	Robust to viewing angles
	Template	HOG [43]	Robust to illumination and shadowing changes, sensitive to object orientations
	Combination	High dimensional composed receptive field histograms [44]	Robust to illumination and minor scene changes
		GIST [45]	Robust to occlusion and viewpoint changes, noise in isolated regions is ignored

Inspired by the application of Haar-like feature in human face detection [41], a small number of objects were first recognized as landmarks in [46]. The recognized objects were then applied as supplementaries to the geometrical features in order to distinguish rooms that had similar geometrical structure and could only be further identified by the objects found there. The Haar wavelets presents the average intensity differences between regions, and likewise can be used to compare the differences between the sum of pixel values in bounding boxes, which allows relatively high robustness to illumination changes.

Ulrich and Nourbakhshthus implemented colour histograms for place recognition by comparing query images with limited images of an entire dataset [47]. Applying colour histograms for image matching was first conducted by Swain and Ballard [42]. The number of colour histograms is based on the number of the colour bands used, e.g. red, green and blue. Each histogram is built by simply counting the number of pixels with a specific intensity in different colour bands. Such feature is robust to viewing angle changes in the case when properties in the environment remain fixed. Furthermore, it provides a highly compact representation of an image and thus requires less memory space. However, it fails to describe spatial relations, which limits its applicability. Filliat *et al.* also adopted this feature to discriminate identical chairs of different colours [48].

Spatial information such as feature location was not included by the holistic methods presented above due to the lack of a segmentation step. Some features divide an image into small discrete regions and then one can compute the global statistics within individual regions in order to obtain some rough spatial information. Grimmett *et al.* used the Histogram of Oriented Gradient descriptor (HOG) [43] to represent training data in order to detect parking space [49]. HOG describes objects by concatenating histograms of the gradient directions computed from the pixels within individual regions divided from an image, called cells. Each histogram is then contrast-normalized across a group of cells, called a block, to decrease the susceptibility to illumination or shadowing changes, except for object orientations. Such feature has a high performance for pedestrian detection if they maintain a roughly upright position.

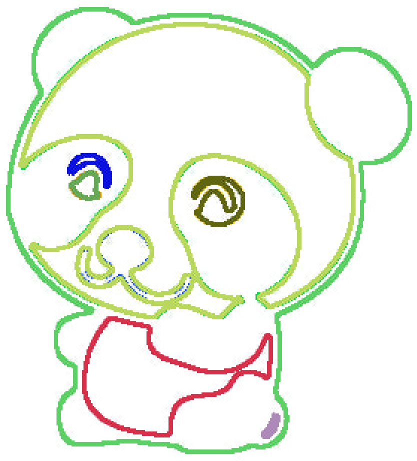
Global features have also been combined in some systems to provide richer representations of the environment. A high dimensional composed receptive field histogram was applied in [44], which consists of normalized Gaussian derivatives, differential invariants and chromatic cues. Siagian *et al.* attempted to incorporate context using the GIST descriptor [45] for scene classification [50, 51]. Orientation, colour and intensity channels are employed by GIST to filter input images with Gabor filters at multiple spatial scales to extract the gist of images. Hinterstoisser *et al.* presented another 3D feature as a complement for DOT feature, named LINE-MOD, by computing object surface normal with a depth sensor [52]. These methods tend to be relatively more robust than using a single global feature since the random noise produced by individual features can be averaged out.

2.2.2 Local Features

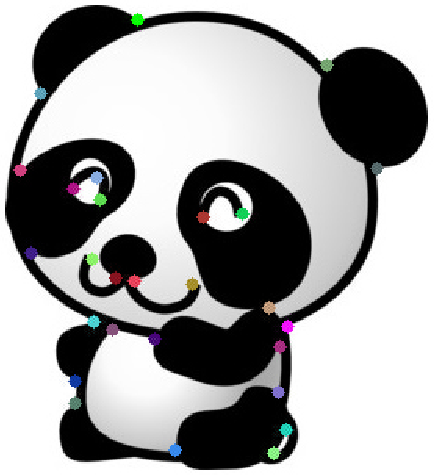
Local features that are widely used in semantic mapping systems for object and place recognition can be further divided into three categories: edge, corner and blob based approaches [53]. Figure 2.4 shows the definition of local visual features in computer vision. An edge is a set of pixels with strong gradient magnitudes or located where the image intensities change sharply. This normally refers to the boundaries between distinguishable regions. A corner is a pixel at which two edges intersect or has edges with two or more directions in the neighborhood. The term corner is additionally used in some cases which differ from our common sense, e.g. a small white spot (corner) on black background, since apart from relying on explicit edge detection, a corner can also be computed from the curvature in image gradient. A blob is a group of connected pixels with similar characteristics. It refers to an interest point as well because many interest point detection methods are essentially based on corner detection at multiple scales.

In this section, the local features are presented accordingly. Table 2.2 presents the differ-

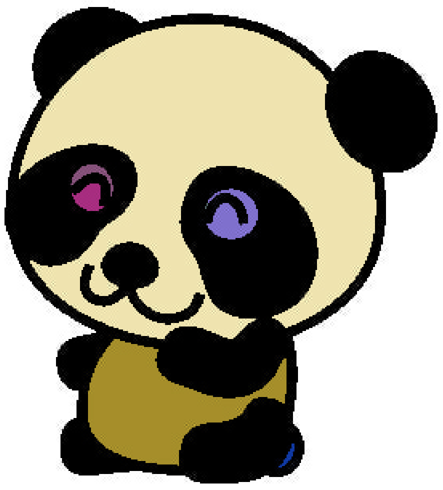
(a) Original image.



(b) Edge.



(c) Corner.



(d) Blob.

Figure 2.4: Definition of local visual features in computer vision.

ences and similarities of the local features used for object recognition in semantic mapping systems.

Edge Based Approaches

The primary characteristic of edges in an image is the sharp change, which is commonly used and captured by classical differentiation based edge detectors. Currently, such edge detectors are only used to generate fundamental cues to construct more robust features or provide complementary information for semantic mapping systems [67]. Ranganathan and Dellaert

Table 2.2: Local features for object recognition in semantic mapping systems.

			Feature	Performance
Detector	Edge	Differen-tiation	Sobel	Computationally efficient, high error rate
			Canny [54]	High accuracy, computationally expensive
	Corner	Gradient	Harris [55]	Sensitive to noise, computationally expensive
			KLT [56]	Computationally efficient, sensitive to noise
		Template	FAST [57]	Computationally efficient, low level of generality
			AGAST [58]	High level of generality, computationally efficient
	Blob	PDE	CenSurE [59]	High accuracy, computationally efficient
		Intensity	MSER [60]	Robust to affine transformations, computationally efficient
Descriptor	Blob	PDE	SIFT [61]	Robust to scale and transformation changes, computationally expensive
			SURF [62]	Robust to scale and transformation changes, computationally efficient
		Template	BRIEF [63]	Computationally efficient, sensitive to viewpoint rotations
			ORB [64]	Computationally efficient, robust to viewpoint rotations
			BRISK [65]	Robust to scale changes, computationally efficient
			FREAK [66]	Computationally efficient, robust to scale changes

[68] converted each training image to a set of regions of interest with Canny edge detector [54]. Clustered edges were obtained to facilitate modelling texture-less objects like desks. The Canny edge detector sets three general criteria for edge detection: low error rate, precise localization on the centre of edges and a given edge in an image should only be marked once. Owing to these criteria, it is one of the most strictly defined methods that provides robust and reliable edge detection. Wu *et al.* attempted to filter each input image with Sobel operator beforehand since they were interested in the spatial structure property of an image rather than detailed textural information [25].

In recent years, edge detection in computer vision have been extended to a broader concept, which is quite similar to object segmentation, named boundary detection. Boundary detection considers an object as a whole. It suppresses the internal edges extracted from the textures within objects and only presents the edges between objects and background. Multiple low-level features are combined to detect boundaries based on machine learning algorithms. However, simply using 2D images tend to be computationally more expensive or less reliable than applying an additional depth channel for them, since it is relatively straightforward to obtain object boundaries from a 3D image. Thus boundary detection using only 2D images is rarely implemented in semantic mapping.

Corner Based Approaches

Primitive corner based approaches rely on gradient assessment, which is a theoretical concept abstracted from our common sense understanding for the term corner. In [68, 69], Harris corner detector [55] was used to facilitate training a database and compute the differential of autocorrelation according to directions directly. A similar detector named Kanade-Lucas-Tomasi (KLT) [56] was employed in [70] for efficient and continuous tracking. Compared to Harris detector, KLT has an additional greedy corner selection criterion, thus is computationally more efficient. However, these corner detectors are not reliable in all circumstances during semantic mapping since gradient assessment method is highly sensitive to noise.

In order to decrease the complexity of gradient assessment and increase computational efficiency, some methods based on template have been implemented. Such features extract corners by comparing the intensity of a pixel with other pixels in the local neighborhood, i.e. a predefined template. Henry *et al.* [71] and Gálvez-López *et al.* [72] attempted to apply Features from Accelerated Segment Test (FAST) [57] for indoor mapping and loop closure, respectively. Based on machine learning algorithms, FAST yields a large speed increase, thus is widely employed by real-time systems. FAST uses a circular template of 16 pixels to evaluate whether a candidate pixel is actually a corner. The candidate pixel is classified as a corner in cases when a certain number of contiguous pixels in the circle are all brighter than the intensity of the candidate pixel plus a threshold value or all darker than the intensity of the candidate pixel minus a threshold value. During the high-speed test for rejecting non-corner points, a decision tree is applied to address the correct rules of the chosen detector. However, FAST suffers from a low level of generality, since it has to be trained for specific scenes before applied. FAST-ER [73] and Adaptive and Generic Accelerated Segment Test (AGAST) [58] increase the performance of FAST in terms of repeatability and generality, by widening the thickness of the Bresenham's circle and training a set of decision trees rather than relying on one tree, respectively.

Blob Based Approaches

Blob based approaches rely on identifying the unique regions in an image by comparing local properties (e.g. intensity and colour) to their neighboring regions. In a blob, specific properties of all the points remain constant or approximately constant, i.e. to some extent, the points are similar to each other. Blob based approaches can be further divided into two categories: keypoint and interest region based approaches. Keypoint based approaches are focused on finding local extrema in scale spaces, whereas interest region based approaches aim at segmenting regions. A scale space is a representation of gradually smoothed images obtained by the rules that can describe basic properties of interest. The scale space presents an image with an additional third dimension. Note that a corner can also be regarded as a keypoint at a specific scale.

Classical interest point based methods are based on Partial Differential Equations (PDE), among which the Laplacian of Gaussian (LoG) is one of most widely used methods. An input image is first convolved by a Gaussian function at a certain scale to represent the image in a Gaussian scale space. The Laplace operator is then applied to obtain strong responses for bright and dark blobs. Compared with LoG, the Difference of Gaussians (DoG) computes the Laplacian of Gaussian operator by the difference between two continuous images smoothed by Gaussian function. DoG can also be viewed as an approximation of the Laplacian operator, thus is computationally more efficient. A hybrid blob detector Hessian-Laplacian combining the Laplacian with the Determinant of the Hessian (DoH) blob detectors has also been proposed, where spatial selection is done by the determinant of the Hessian matrix and scale selection is performed with the scale-normalized Laplacian.

Based on DoG and Hessian matrix, Lowe proposed Scale Invariant Feature Transform (SIFT) [61], which was widely applied by robot SLAM and object recognition systems [29, 74, 75, 76]. The original image is convolved with DoG to identify potential interest points that are invariant to scale and orientation changes. The points selected from the training image which usually lie on high-contrast regions of images such as edges and corners are detectable even under changes in image scale, noise and illumination. Another property of these points is that the relative positions between them in the original image remain stable from one image to another. Subsequently, low contrast and unstable points are rejected based on their locations and scales. Orientations are then assigned to the points based on gradient directions, thus providing invariance to transformations. Finally, SIFT computes a descriptor vector (histogram of oriented gradient) as a representation for each keypoint. Compared with other feature descriptors, SIFT is highly robust to scale and transformation changes, but is computationally expensive. A refinement of SIFT was proposed by Mikolajczyk and Schmid named Gradient Location and Orientation Histogram (GLOH) [77], which proves to be more distinctive than SIFT, yet requires even more computational cost.

Riazuelo *et al.* initially extracted Speeded Up Robust Features (SURF) from each image and stored them for latter object recognition in the RoboEarth database, which is a knowledge-based system providing web and cloud services [78]. SURF has claimed to be several times faster than SIFT and its accuracy remains relatively acceptable. SURF employs integral images and uses square-shaped filters to approximate the determinant of Hessian matrix during Gaussian smoothing, thus is more computational efficient. Morisset *et al.* used Centre Surround Extrema (CenSurE) [59] to obtain a visual odometer in real-time [79]. CenSurE is another approximation of LoG. Compared with SIFT and SURF, CenSurE features are evaluated for all the pixels across all scales in raw images. This leads to higher accuracy. Moreover, even seeking extrema at all scales, it still maintains a relatively low computational cost by adopting a set of simple centre-surround filters. Implementations of these refinements in semantic mapping systems can also be found in [48, 80, 81].

Due to the high demand for real-time applications, Gálvez-López and Tardós adopted Binary Robust Independent Elementary Features (BRIEF) [63] to find the best frame-to-frame matches for real-time localization over long periods [72]. BRIEF is a binary string constructed by classifying image patches according to pairwise intensity comparisons, which leads to a small memory usage and is highly computational efficient during recognition.

Inspired by FAST and BRIEF corner detector based on template, Rublee *et al.* presented Oriented FAST and Rotated BRIEF (ORB) by estimating the patch orientation [64], thus is invariant to viewpoint rotations. The scale pyramid is also applied to increase its robustness to scale changes. Such method was employed in [82] to generate photometric feature for RGB-D mapping. Grimmett *et al.* used Binary Robust Invariant Scalable Keypoints (BRISK) [65] to build maps for automated parking [49]. BRISK applies AGAST detector in both image plane and scale space to classify keypoints so that it is invariant to scale changes. A keypoint detector motivated by human retina and derived from BRISK was presented by Alahi *et al.* [66], named Fast Retina Keypoint (FREAK), which was applied in [83] for facial point detection and emotion recognition. Compared to BRISK, FREAK has a higher density of points near the centre of the sampling grid.

Meger *et al.* [74] and Sun *et al.* [84] applied Maximally Stable Extremal Region (MSER) [60] in their systems to provide object location information for an attentive system and to extract lane marking features, respectively. MSER is one of the most widely used methods for interest region detection. It is robust to affine transformations and is highly computationally efficient. However, it is sensitive to image blurry changes. Moreover, MSER is a region detector in essence, thus is only suitable to distinguish objects with little variation in colour from high-contrast scenes.

Discussion

One of the most important factors in evaluating the feature detectors and descriptors implemented in semantic mapping systems is their accuracy (reliability). To assess it, a repeatability criterion presented by Schmid *et al.* measures whether or not the same feature is detected in two or more different images of the same scene under varying viewing conditions [85]. The repeatability is a ratio between the accurate pairing number and the minimum number of keypoints detected in the given images. The repeatability of some local features is shown in [38]. Three image transformations are considered: zoom factor, viewpoint and rotation.

With respect to some SLAM or object recognition systems running in real-time, the computational complexity of the applied feature detectors also plays a key role. Canclini *et al.* [38] also evaluated the efficiency of some widely used keypoint detectors in semantic mapping systems. More specifically, the average processing time was assessed based on the number of keypoints detected. FAST and AGAST are computationally more efficient than other detectors.

Another noticeable difference is that the processing time of CenSurE and ORB remains constant, whereas with the increase of the keypoint number, the processing time of other detectors grows linearly. The influence of image scale changes on the detectors was also presented. The processing time for all the detectors raises as a quadratic function with the increase of image scale. Again, SIFT and SURF are several times more time-consuming.

2.3 Recognition Approaches

This section presents some basic object/place recognition approaches in semantic mapping systems, namely global, local and hybrid approaches [86]. Object recognition methods based on global features are classified into global approaches. Such approaches also consist of some place recognition methods which employ image segmentation algorithms directly rather than referring to the properties in the environment. Local approaches include pixel-wise operations on the basis of local feature extraction and the straightforward sampling of pixel contributions. Some systems combine global approaches with local approaches to achieve a more robust performance, which is discussed in hybrid approaches. In addition, information that is retrieved to distinguish individual instances within an object class (e.g. shampoo or conditioner, someone's office) is also discussed. We finally summarizes some recent deep learning based approaches.

2.3.1 Global Approaches

Based on the global statistic features retrieved from texture, Hidden Markov Model (HMM) was applied in [87] for place recognition and new place categorization. For HMM, the states which represent different locations are not directly visible, whereas the output acquired from the states is visible. Compared by using a uniform transition matrix, HMM provides a significant increase in recognition performance. Furthermore, the computational cost is quite low and can be neglected during mapping. However, it is only applicable for a small database. Mozos *et al.* implemented a cascade of classifiers [88] which depended on boosting to detect 8 different objects in order to recognize 6 places [46]. Boosting is a supervised learning-based method combing several simple weak classifiers to achieve a relatively high performance. For each of the weak classifiers used, the requirement is that its accuracy should be better than random guessing. The accuracy of the weak classifiers leads to their distributions once they are added. The cascade of classifiers is essentially a degenerated decision tree which rejects non-object regions at each stage and retains interest regions for further classification. Although the training time is long, the prediction can be run in real-time. Gentle AdaBoost was also applied by Murphy *et al.* for object and scene recognition [45].

In the case of colour histogram features, Ulrich and Nourbakhsh used a simple unanimous voting scheme to classify places [47]. The input images were voted by each colour band with the smallest minimum matching distance. A certain place was classified when the colour bands unanimously voted for the same place and the total confidence was above a threshold. Such a method is quite straightforward and computationally efficient. However, one important prerequisite is that the visible properties in scenes should remain relatively fixed and its performance drops when it comes to a large database (over 100 images).

Support Vector Networks (SVM) [89] was applied with HOG feature by Dalal and Triggs [43]. SVM is a set of supervised models with associated learning algorithms widely used for data analysis and pattern recognition. The training process tries to build a model by the given examples to assign new examples into two categories, making it a non-probabilistic binary linear classifier. This step is essentially a process to find a model with high performance, i.e. a clear gap that is as wide as possible. In [43], positive examples (images which contained pedestrians) and negative examples (person-free images) were provided for SVM training. The implementation of linear SVM rather than using a kernel decreased the computational cost of the system. Pronobis *et al.* also applied SVM to recognize places [44] based on a kernel [90], which proved to achieve better performance for histogram-like features. Results in this paper showed that the places were recognized with high precision and robustness even when training on images from one camera device and testing on another. Inspired by Taylor and Drummond [91], the Streaming SIMD Extensions (SSE) were applied to efficiently compute error functions [92].

2.3.2 Local Approaches

Apart from the approaches mentioned above, one of the most promising works has been done by Lowe [61]. Once the SIFT features are detected and described, recognizing objects becomes a problem of finding groups of similar descriptions that have all undergone the same transformation. More specifically, an interest point in the test image is compared to an interest point in the reference image by the differences between their description vectors, which is based on

Euclidean distance. For rapid computation against large databases, the features are put in a KD-tree, which is a data structure based on nearest neighbor searching for large databases. The Hough transform is used to cluster the features that belong to the same object. Clusters of at least 3 features that agree on the object and its pose are identified as candidate matches. A least-square approximation is then made to obtain the best estimated affine projection parameters, which are further applied to decide whether to keep or reject the matches. This method has been widely implemented in robot semantic mapping systems [75, 74, 29] thanks to its high robustness. However, due to the complexity of the SIFT feature, the recognition process still suffers from high computational cost.

In the case of the binary features inspired by modern computer architectures, such as BRIEF, BRISK, ORB and FREAK, the Hamming distance is used for matching. The Hamming distance between two feature vectors is the number of positions at which the corresponding symbols are different. Such matching method is highly computationally efficient. However, the accuracy is lower than the method presented by Lowe.

2.3.3 Hybrid Approaches

Some systems adopted global features, local features and depth information to generate a more robust recognition performance. Depth information additionally provides spatial dimensions of objects and represents objects in more detail, thus leads to a higher recognition accuracy compared to using solely 2D features. Histograms of Oriented Energy and colour were directly applied for object detection in [93]. Stückler *et al.* employed region features in both colour and depth space and applied object-class segmentation algorithms for semantic mapping [94], based on Random decision Forests (RFs), which is an ensemble learning method for classification and has been demonstrated to achieve comparable performance to SVM [95]. In this work, a subset of images from the training set was randomly selected as a sample to train the decision trees. Small objects were better sampled for training, thus the actual individual distributions of class labels were reassigned according to this. One advantage of RFs is the high computational efficiency during outputting, yet the training time is still relatively long.

Filliat [96] and Martínez-Gómez *et al.* [97] employed a Bag of Binary Words (BoW) model [98] to incrementally learn to recognize different rooms from any robot position. BoW is inspired by a technique in document classification and consists of two phases, namely representation (indexing) and recognition (retrieval). Image features that are robust to intensity, rotation, scale and affine are detected and described by independent feature descriptors with vectors, such as SURF and FAST (SIFT, colour histograms and normalized gray level histogram in this paper). Subsequently, the vectors are clustered by vector quantization algorithms, e.g. K-means clustering [99]. The predefined codewords (words in documents) are then assigned to the clusters to generate a codebook (a word dictionary), thus the images are represented by a histogram of codewords. In the case of the recognition stage, generative or discriminative models such as Naive Bayes classifier, SVM and AdaBoost are applied as the classifiers. Such a method is quite flexible in terms of both applicable features and recognition approaches. However, the spatial relationships among the clusters are ignored when BoW is used alone, which has been compensated by Lazebnik *et al.* [18].

Text or signs can provide location information directly. Most text recognition systems implemented Optical Character Recognition (OCR) for classification [100, 101, 102], which is an off-the-shelf technology to convert images of typed, handwritten or printed text into machine-encoded text. Sami *et al.* [101] adopted a back-projection of the colour histogram to locate interest regions and applied Canny edge detector to remove background. A pan/tilt/zoom camera was used in [102] to provide better focusing performance on potential text regions in the wild. However, text retrieval still suffers from low performance in cluttered environments, which limits its practicability.

2.3.4 Deep Learning Based Approaches

Convolutional Neural Networks (CNNs) have recently been widely used as robust visual feature extractors in the computer vision and machine learning domain and have shown better performance in terms of changing environments, viewpoints, lighting conditions, objects, etc. [103, 104]. Sharif Razavian *et al.* [104] have shown that CNNs outperform BoW in most recognition tasks in terms of large datasets.

Although most CNNs are trained for object recognition, some researchers have managed to modify these models for other related but different tasks such as place recognition and object detection [104, 105, 106] since the generic features learned by different models from holistic images in different datasets are versatile and transferable [107, 108].

Moveover, the descriptive features extracted and the extremely large and diverse data used for training also benefit visual odometry. PoseNet proposed by Kendall *et al.* [109] shows the first implementation on pose estimation, which directly generates the six degrees of freedom (6-DoF) of an camera from a single RGB input image. The model GoogLeNet [110] pre-trained on other classification tasks is leveraged for pose regression. The softmax layers that originally output classification results are removed and replaced by a seven dimensional pose vector. The last fully connected layers are also modified. CNNs extract more robust features than traditional feature detectors and achieve high accuracy even when extreme conditions exist, such as intense lighting and blurry images. PoseNet can also be easily generalized to other scenes through transfer learning technique. The model on the new task can thus be trained with smaller dataset and shorter time. Li *et al.* [111] incorporated another CNN stream to PoseNet and fed depth images into this stream to enhance the relocalization accuracy. ORB-SLAM is used to label the collected images as ground truth. Recurrent Convolutional Neural Networks [112] were also employed by Wang *et al.* [113] for pose estimation. However, all of these deep learning based methods require ground truth poses for training, which can be quite expensive and labour-intensive to produce.

2.4 Semantic Representation

Semantic representation is the interpretation process from objects or places to a human-compatible or even human-like language. Some systems presented above apply classification or segmentation methods for the purpose of recognizing specific objects or scenes, thus the semantic information is directly obtained. In this section, we mainly focus on the semantic information inferred by robots.

Early systems [37, 29] adopted the idea that a semantic map consists of two separate but tightly interconnected parts: a spatial part and a terminological part [114]. This is a typical structure of hybrid Knowledge Representation (KR) systems [115], as shown in Figure 2.5. The spatial part contains raw images from the sensors, geometric information of the environment and connectivity between the rooms, whereas the terminological part consists of general semantic knowledge about the environment, giving meanings to the features of the corresponding properties in the environment in terms of general concepts and relations. These two hierarchies are interrelated by the concept of anchoring [116]. In [37], the NeoClassic AI language was employed to establish the conceptual hierarchy and provided the robot with inference capability. However, the conceptual knowledge was hand-coded into the system and uncertainties about the properties in the environment were not included into the representation.

Vasudevan and Siegwart attempted to classify places based on objects as well [117], and their system was fully probabilistic. In their system, objects were grouped into predefined clusters and conceptualised during the training process. A simple Naive Bayesian Classifier (NBC) was then employed to infer and identify the place categories on the basis of the clusters.

Meger *et al.* [74] developed an attentive system projecting the location and semantic information of the recognized objects back into the grid map. Since the object recognition subsystem was trained by collecting object model data through submitting text-based queries to internet image search engines, the semantic information was thus easily incorporated into the object models. Once an object was observed, the semantic information was directly acquired. A similar work [118] built the spatial-semantic object models based on the LabelMe database [119].

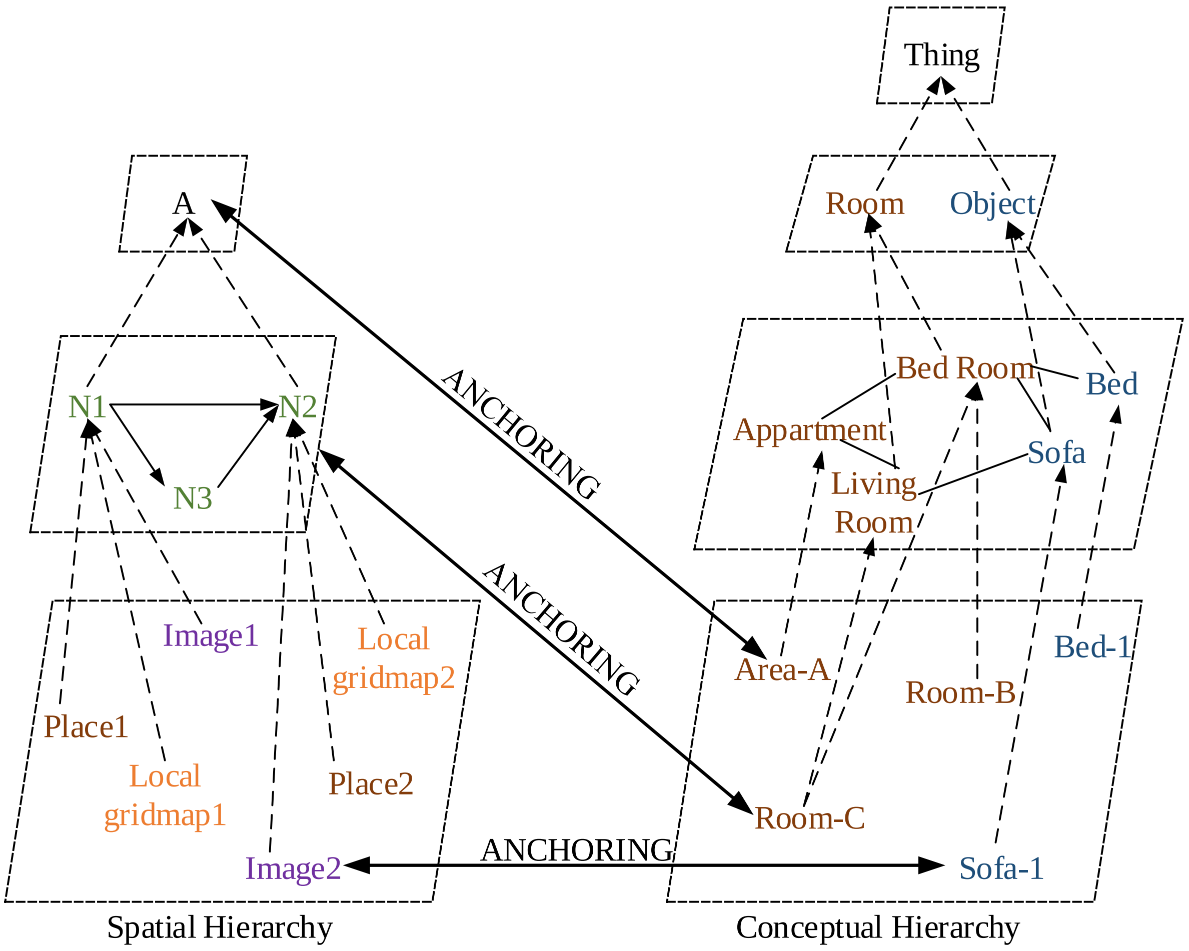


Figure 2.5: The spatial and conceptual hierarchy interrelated by anchoring.

Bayes' theorem was applied to define place categories by the recognized objects. More specifically, a cluster model grouped objects on the basis of their locations, thus the place categories were just another representation of the clusters.

Zender *et al.* [75] and Capobianco *et al.* [120] encoded conceptual knowledge into an Web Ontology Language-Description Logic (OWL-DL) ontology. In [75], a description-logic reasoner employed some situated dialogues between a robot and an user to provide new knowledge for the robot to further infer. In addition, a laser sensor was implemented for place classification. More specifically, a navigation node (a marker) was placed in the metric map after the robot moved 1 metre away from the last node. The nodes were then connected and classified for room type identification. Pronobis *et al.* extended such method and applied the doors detected in indoor environments to bound areas [3]. The final map is shown in Figure 2.6. In [120], a standard methodology for representing and evaluating semantic maps was proposed. The formalisation consisted of a reference frame, spatial information and a set of logic predicates. With this system structure, the performance of semantic representations can then be compared against those of other systems.

2.5 Typical Applications

Service robots are gradually working their way into our daily lives to become household servants, healthcare systems and even cognitive companions. The primary responsibility of service robots is to obey the orders given by humans and perform tasks with high efficiency and accuracy. A semantic map provides a friendly human-robot interface and enables these service robots to be used by general public without the need of training.

2.5.1 Indoor Applications

Some applications can be found in indoor environments. Galindo *et al.* presented a typical autonomous navigation method based on a pre-built semantic map [37]. In their experiment, the robot was given a command “go to the bathroom”. Following this command, the robot inference system found a node in the topological map and the spatial information in the metric map that connected to the node was retrieved by anchoring. Thus, the command was translated to “go to the node” and then executed between the topological and metric hierarchies. Figure 2.7 describes this navigation method.

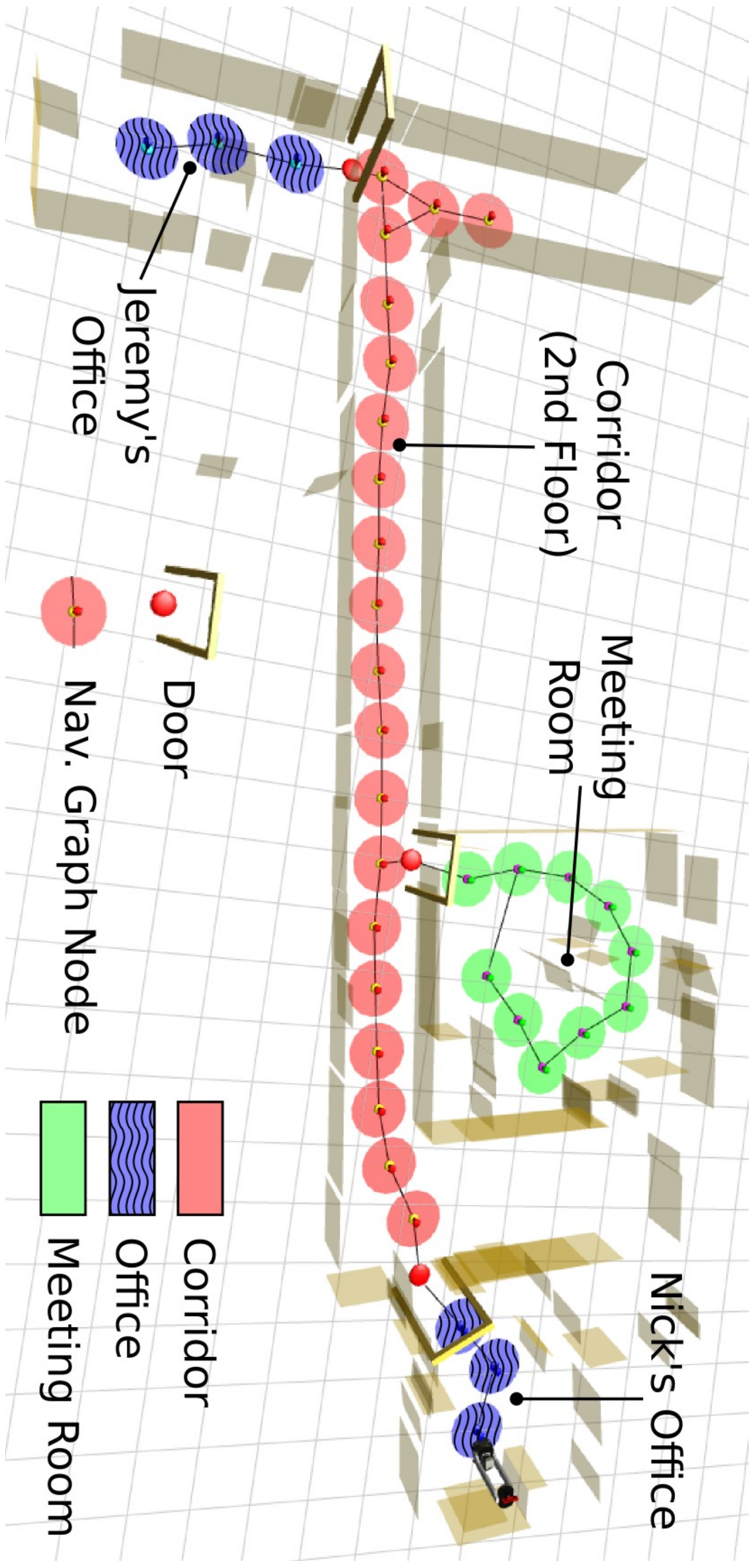


Figure 2.6: The final semantic map obtained by Pronobis *et al.* [3]

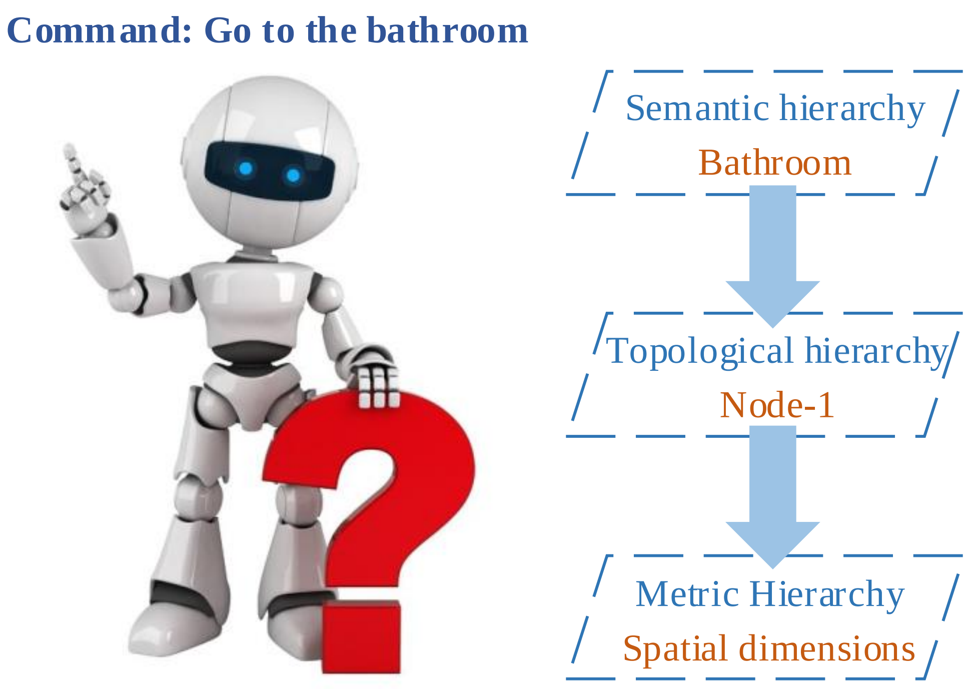


Figure 2.7: Robot navigation based on semantic map.

The authors in [121] enhanced the inference capability of a robot with a semantic map. The common knowledge known by almost everyone was applied to detect deviations from normal conditions. A goal was then autonomously generated by the encoded information about how things should be, e.g. if a bottle of milk was observed on a table, the robot would set a goal by itself and bring the milk into a fridge. Crespo *et al.* also presented a reasoning module to infer new knowledge for mobile robots [122]. A relational database used for storing objects and perceiving information was implemented to provide inference capability based on objects links.

Blodow *et al.* extracted semantic information by a laser scanner and a high-resolution 2D camera for a PR2 robot to perform tasks in kitchen environments [4]. Segmentation algorithms were applied to identify and distinguish certain kitchen facilities and their functional components. The robot was capable of analysing the task-relevant objects, locating them in the map and acting on them, e.g. using the handle to open the drawer and close it, as shown in Figure 2.8.

A semantic map can be inversely utilized to further improve robot localization capabilities. One basic method is to identify room categories by specific objects. For example, the room is classified as a kitchen once an oven is found inside. Initial localization errors can be reduced by

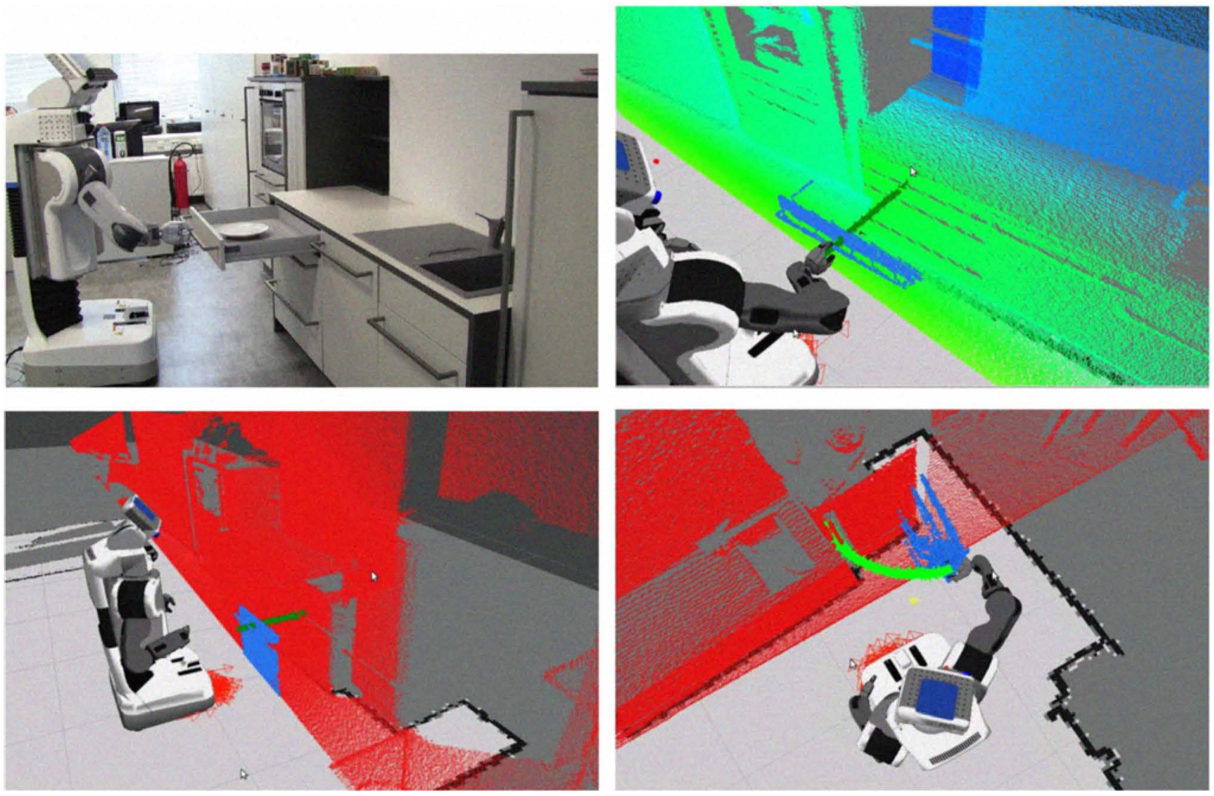


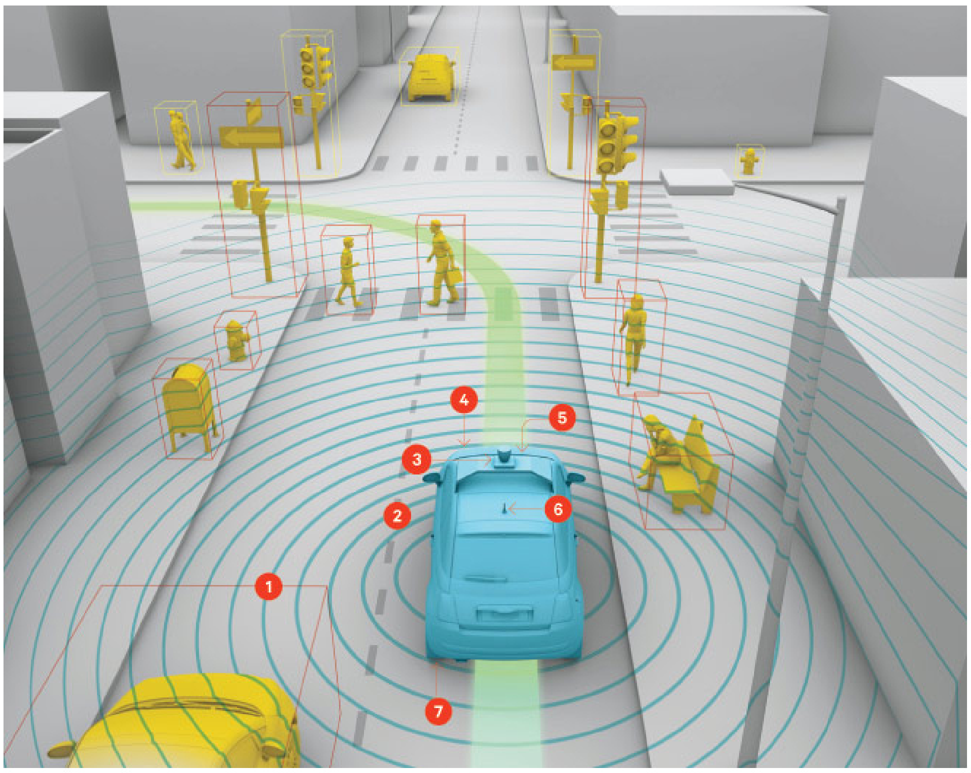
Figure 2.8: Interaction between the robot and the kitchen facilities [4].

reasoning about the expected location of the recognized objects [37]. Ko *et al.* extended such a method by continuously searching for other memorized objects as landmarks before referring to spatial relationships since one object may not be enough to infer the accurate location [123]. In addition, the time for searching regions in topological and metric space were largely reduced by discarding the irrelevant areas. The computational cost in the initial stage during robot localization were thus be minimized [114].

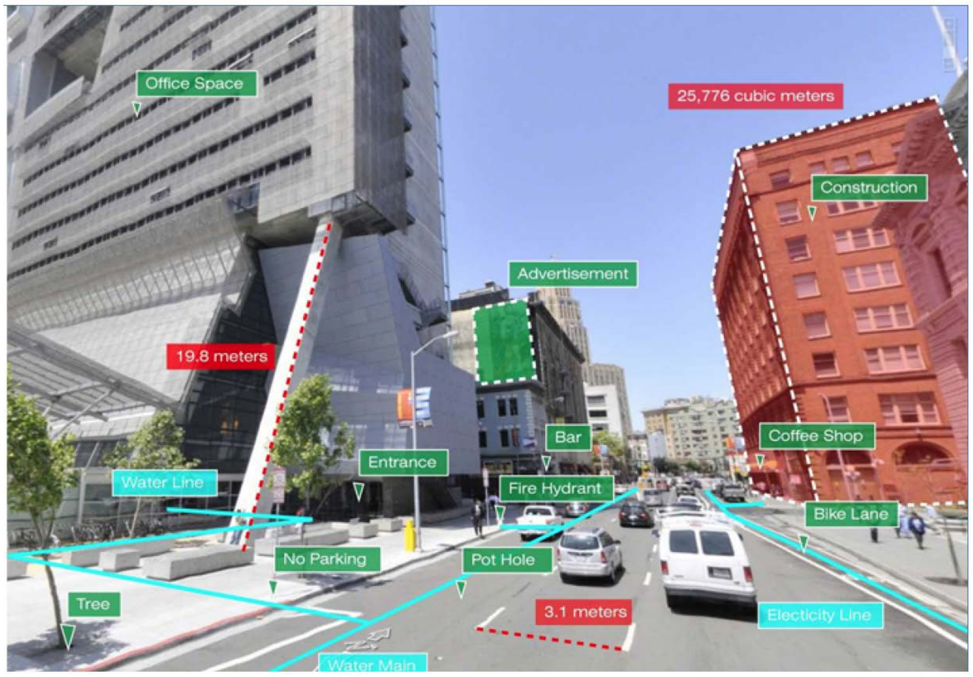
2.5.2 Outdoor Applications

Semantic maps can also be applied to outdoor environments. Wolf and Sukhatme analysed and classified terrain to resolve issues relating to non-navigable areas during path planning [28]. Boularias *et al.* additionally adopted natural language to command a mobile robot for navigation in outdoor environments, e.g. “Navigate to the building behind the pole” [124]. Bernuy and Solar presented a graph based topological semantic mapping method for autonomous off-road driving [125].

Other applications include self-driving cars and augmented reality (AR) as shown in Fig-



(a) Object identified by a self-driving car [126].



(b) Augmented reality technique pointing out the nearest destination [127].

Figure 2.9: Outdoor applications of semantic maps.

ure 2.9. Reasoning about environments with additional semantic information plays a key role for self-driving cars since the cars should be able to recognize roads, pedestrian crossings, humans, traffic lights, etc. However, current self-driving cars still have difficulty in identifying some kinds of objects such as plastic bags which are harmless but causing the vehicles to veer unnecessarily. When a police officer signals them to stop, they may not reacted accordingly. These problems can be solved by associating the metric map with semantic information. With respect to AR, the augmentation should be conventional in semantic context with environmental elements, e.g. directing the way by virtual paths. Semantic maps are necessary for identifying objects and destinations in real environments.

2.6 Summary

Chapter 2 presented a review on current semantic mapping system architectures. The visual sensor based approaches to semantic mapping were first introduced, and the features extracted from images were then detailed. Subsequently, the recognition and classification methods based on the extracted features were discussed, as well as the direct segmentation methods. Deep learning based methods were also introduced. Lastly, the semantic representation strategies and typical applications were presented.

Chapter 3

Building Semantic Maps for Human-Robot Interaction

The traditional environment maps built by robots include both metric ones and topological ones. These maps are navigation-oriented and not adequate for service robots to interact with or serve human users who normally rely on conceptual knowledge or semantic contents of the environment. Therefore, the construction of semantic maps becomes necessary for building an effective human-robot interface for service robots. This chapter aims to build a 3D environment map with an RGB-D sensor and extract semantic information from RGB images to help blind people navigate at home. A novel approach is presented to diagnose and eliminate errors during semantic extraction and loop closure detection.

3.1 Introduction

Nowadays, 285 million people are estimated to be visually impaired worldwide, among which 39 million suffer from total blindness [128]. Guide sticks and dogs have been deployed to lead blind people around various obstacles. However, the guide sticks are too simple to be effectively used and the guide dogs are expensive to be trained. Both of them are unable to interpret street signs or complex outdoor scenery reliably, let alone providing semantic guidance. Thus, it remains a major challenge for blind people to live independently at home.

To build a robotic system to help blind people, a semantic map is the preliminary requirement and can be deployed to guide them around home. Traditional maps, namely geometrical ones and topological ones, are only navigation oriented, which enable mobile robots to navigate around and plan a path for reaching a goal [28]. However, these maps are inadequate for blind people who normally need semantic information interpreted from scenes. Therefore, it is necessary to build a semantic map and provide voice guidance for blind people based on semantic maps, e.g. "There is a chair in front of you" or "You are in the kitchen now".

In a semantic map, nodes representing places and landmarks are named by linguistic words. Examples of these include names and categories of different objects, rooms and locations [21]. More specifically, a semantic map can be regarded as an extension of a hybrid map, which contains geometric description, topological connectivity and semantic interpretation [30]. It provides a friendly way for service robots to communicate with users.

More recently, representing environments using semantic data has become a popular research domain and drawn enormous attentions from different fields. In the early stage of its development, range sensors were widely applied to build 2D projections of scenes or 3D spatial models of the physical environments and then complemented by further semantic information. Nüchter and Hertzberg [30] provided a fast plane extraction method in indoor environments with a 3D laser scanner to distinguish between different architectural components such as ceilings, floors, doors and walls. In [129], laser range data was used to classify rooms, corridors, doorways and hallways in indoor environments. Although these works have undoubtedly promoted the development of semantic mapping, the recognizable objects and locations were restricted to a small scope due to the limited features provided by range sensors. Thus, attentions were drawn to visual sensors which contain richer information than non-visual data. Moreover, the application of visual sensors in semantic mapping seems to be a sensible decision since we humans perceive the world through our eyes. Visual data allows the representation of not only low-level features such as lines, corners and shapes, but also high-level features such as colours, relations and texts. In this way, the additional features lead to a wider variety of objects that can be recognized, which highly enriches semantic maps.

At first, recognizing objects or places was normally done by using conventional cameras that record two-dimensional images. In [130], models of indoor Manhattan scenes were acquired from individual images generated from a 2D camera and then assigned with semantic labels. Tian *et al.* addressed door modeling and detection problem to assist blind people to access unfamiliar indoor environments [131]. Wu *et al.* [25] and Neves dos Santos *et al.* [132] employed

2D visual data to tackle the problem of place recognition in semantic mapping. SLAM and object recognition with monocular cameras were presented by Civera *et al.* [133] and Riazuelo *et al.* [78], respectively. However, these methods only use 2D images to recover scenes, which inevitably lose the absolute scale of the scenes. An additional post-processing step is needed to inject scale into the pre-built maps and recover the true scale in real world. Therefore, some researchers try to seek other solutions to this problem.

In recent few years, mapping environments into 3D point clouds and extracting semantic information from them have become a new trend [5, 134, 135, 136] due to the availability of low-cost and light-weight 3D point cloud capturing devices, such as stereo cameras and RGB-D sensors, which allow the application to small robot platforms or even wearable devices easily. Compared to 2D images, 3D point clouds overcome the limitation in the data-stream itself by providing additional depth data. The absolute scale can thus be easily recovered. Moreover, humans recognize and perceive a 3D world in terms of our eyes, which means recognizing objects in a 3D world tends to be more natural and accurate, especially when it comes to a large variety of goods in our daily life [34]. However, how to extract semantic information and detect loop closure more accurately still remain challenging.

In this experiment, both RGB and depth images are deployed for estimating the sensor poses and then generating 3D geometrical maps. Semantic information is extracted by matching the RGB images with the object images in the database. We have also implemented a simple strategy to detect false positive matches and eliminate recognition errors. The methods are detailed in Section 3.2 and 3.3.

The rest of Chapter 3 is organized as follows. Our 3D geometrical mapping and semantic information extraction methods are detailed in Section 3.2 and 3.3. Subsequently, experimental results are presented and discussed in Section 3.4. A brief conclusion is given in Section 3.5.

3.2 Feature Based Metric Map Building Method

Figure 3.1 presents the block diagram and configuration of the proposed semantic mapping system. The blue blocks are the inputs (RGB images, depth images and database) and the output (semantic map) of our system. The orange ones represent the 3D mapping process and the green ones describe the semantic information extraction process. Algorithm 3.1 presents the order in which the processes are carried out.

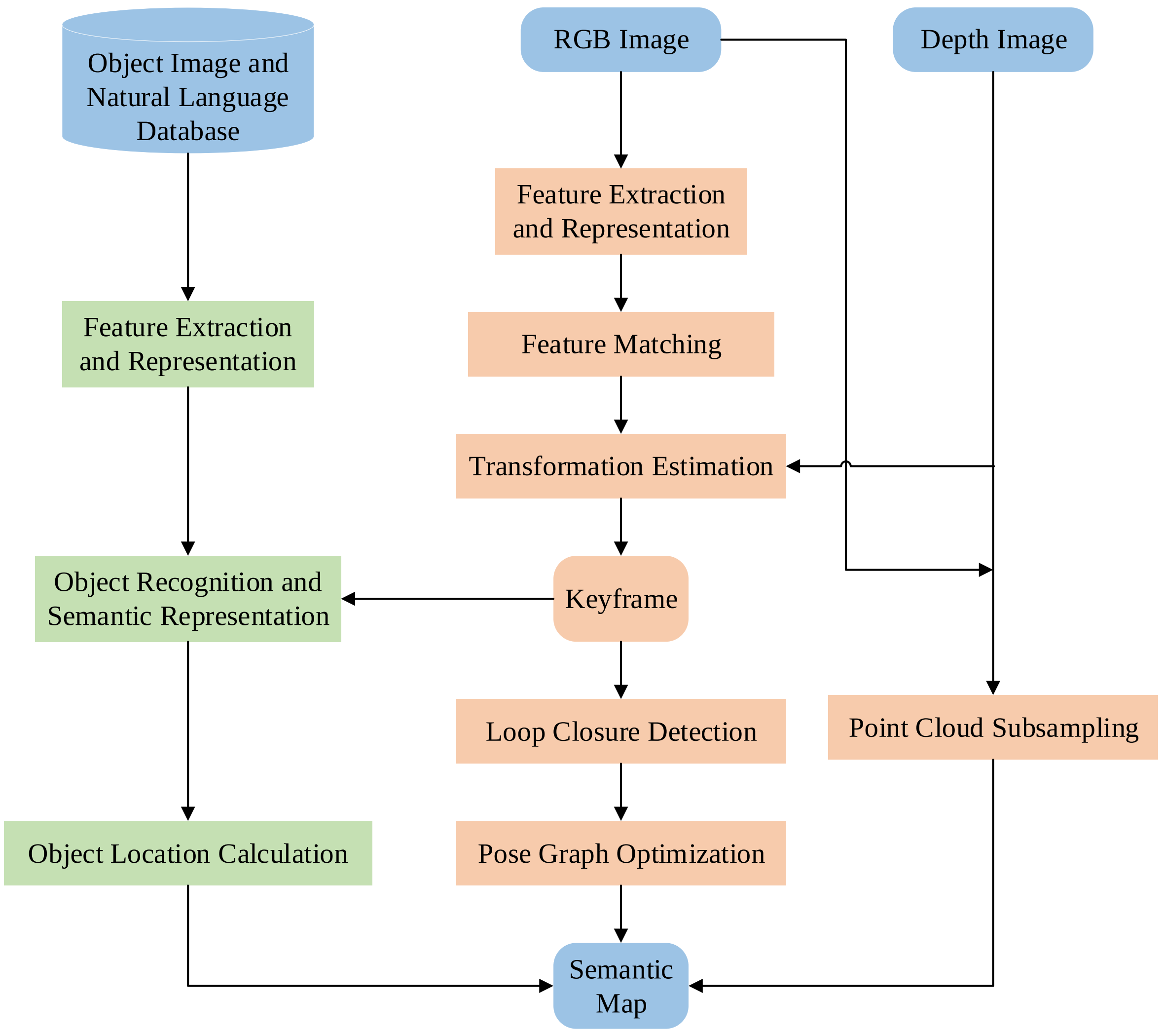

Object Image and Natural Language Database
RGB Image
Depth Image
Feature Extraction and Representation
Feature Extraction and Representation
Feature Matching
Transformation Estimation
Object Recognition and Semantic Representation
Keyframe
Loop Closure Detection
Point Cloud Subsampling
Object Location Calculation
Pose Graph Optimization
Semantic Map

Figure 3.1: System overview. Blue boxes: inputs and output. Orange boxes: mapping process. Green boxes: semantic information extraction process.

As for the mapping process, specific features are detected and feature descriptors are computed and obtained from RGB images. The descriptors are then compared with the ones of the previous key-frame and their 3D coordinates to estimate a rough transformation matrix (rotation and translation). If the transformation is substantial enough, a new key-frame is added. Subsequently, loop closure detection is carried out by matching the current key-frame with some of the previous key-frames. A pose graph is then built and optimized using g2o [137] in order to obtain a relatively precise trajectory.

Finally, the point clouds generated from the input RGB and depth images are down-sampled and projected into a common coordinate frame. A detailed 3D model is thus represented. During semantic information extraction, the images and names of the objects are first fetched from the database. We then apply object recognition in each of the key-frames. Once an object is detected, we calculate its location in the global coordinate and then store it in a file which can be used for further inferring and navigation.

3.2.1 Feature Extraction

In our system, we adopt SURF (Speeded up Robust Features) [62] for both mapping and object recognition. SURF has claimed to be several times faster than SIFT, and the accuracy still remains relatively acceptable. SURF employs integral images and square-shaped filters to approximate the determinant of the Hessian matrix during Gaussian smoothing, thus can be computational efficient.

We have also tested SIFT (Scale Invariant Feature Transform) and ORB (Oriented FAST and Rotated BRIEF) features. SIFT tends to be more computationally expensive, however the absolute trajectory error (ATE) remains almost the same. Computing an ORB feature is several times computationally efficient than SURF, but the number of good matches that provide correct transformation estimation (inliers) is always not enough. Therefore, we finally decided to employ SURF in our system.

3.2.2 Transformation Estimation

Due to the distinctive visual features in indoor environments, the motion of the sensor can be estimated by measuring the similarity or the distance between the feature descriptors extracted from two sequential key-frames. In this experiment, we apply Perspective-n-Point (PnP) to solve this problem, which originates from camera calibration. PnP solves the problem of estimating the pose of a calibrated camera given a set of n points in the 3D world and their corresponding 2D projections in a image. The sensor motion which consists of 6 degrees of freedom (DOF) can thus be estimated and represented by a rotation matrix (roll, pitch and yaw) and a translation matrix.

Assume F_p and F_q are two RGB images with N pairs of matched points p_i and q_i,

$$p = \{p_1, p_2, ..., p_i, ..., p_N\} \in F_p, \tag{3.1}$$

$$q = \{q_1, q_2, ..., q_i, ..., q_N\} \in F_q. \tag{3.2}$$

Algorithm 3.1: Feature based semantic mapping.

```
Input  : Consecutive RGB images $\{I_1, I_2, ..., I_N\}$
         Associated depth images $\{D_1, D_2, ..., D_N\}$
         Object images in the database $\{O_1, O_2, ..., O_M\}$
Output: Semantic map
function generate_Global_Pose_Graph
    take $I_1$ as the first key-frame
    g2o initialization
    for $(i = 1; i < N; i++)$ do
        compute SURF feature $V_i$ from $I_i$
        match $V_i$ to the previous key-frame using FLANN
        calculate $min\_Matching\_Distance$
        $thres\_Distance = 4 \times min\_Matching\_Distance$
        if distance<thres_Distance then
            $good\_Match++$
        end
        set $thres\_Good\_Match\_Number \in [10, 20]$ based on experimental experience
        if good_Match<thres_Good_Match_Number then
            continue
        end
        compute point cloud $C_{i+1}$ using $I_{i+1}$ and $D_{i+1}$
        compute transformation $T$ using $I_i$ and $C_{i+1}$
        set $thres\_Transformation = 0.3$ based on experimental experience
        if T>thres_Transformation then
            set $I_i$ as a new key-frame
            add a new node and edge to g2o
            while $j < M$ do
                match $V_i$ to images in the object database
                if $neighbouring\_Good\_Match\_Number\_Verification = true$ then
                    add label to $I_i$
                end
            end
            if $loop\_Detection = true$ then
                if $neighbouring\_Good\_Match\_Number\_Verification = true$ then
                    add a new node and edge to g2o
                end
            end
        else
            continue
        end
    end
end
global pose graph optimization using g2o
subsample $C_k$ and join $C_k$ together using PCL
```

$Q = \{Q_1, Q_2, ..., Q_i, ..., Q_N\} \in F_Q$ are the associated 3D points of p. The 3D point Q_i can be calculated from q_i by

$$\begin{cases} x_{Qi} = (u_{qi} - c_x) * z/f_x \\ y_{Qi} = (v_{qi} - c_y) * z/f_y \\ z = d_{qi}/s \end{cases}, \tag{3.3}$$

where (x_{Qi}, y_{Qi}, z_{Qi}) are the 3D coordinates of Q_i, (u_{qi}, v_{qi}) are the 2D coordinates of q_i, f_x, f_y are the focal lengths expressed in pixel units, (c_x, c_y) is the principal point that is usually at the image centre, s is a scale factor, d_{qi} is the depth reading of point q_i. We can then estimate the camera transformation matrix using

$$sp_i = K \left[\hat{R}_{P,Q} \middle| \hat{T}_{P,Q} \right] Q_i \tag{3.4}$$

or

$$s \begin{bmatrix} u_{pi} \\ v_{pi} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_{Qi} \\ y_{Qi} \\ z_{Qi} \\ 1 \end{bmatrix} \tag{3.5}$$

where $\hat{R}_{P,Q}, \hat{T}_{P,Q}$ are the expected rotation and translation matrices, respectively.

Although four pairs of points are enough to estimate the transformation, some possible errors could appear during matching and may affect the result. Therefore, we apply RANSAC to retain good matches that provide correct estimation and improve the robustness in terms of outliers. A threshold is set for allowed distance between the observed and computed point projections in order to verify inliers. The iteration will stop if the RANSAC algorithm at some stage finds certain number of inliers. Particularly, we minimize the sum of squared distances between two frames

$$\min_{\hat{R},\hat{T}} \sum_{i=1}^{N} ||P_i - (\hat{R}Q_i + \hat{T})||_2 \tag{3.6}$$

in order to obtain the accurate transformation matrix. If the transformation is substantial enough, the current frame is regarded as a new key-frame.

3.2.3 Loop Closure Detection

A global pose graph can be generated by the transformation estimation process discussed above. However, the rough individual estimations between pairs of consecutive key-frames are noisy, especially when few features are detected. Loop closure is thus applied to reduce the accumulated noise and increase the mapping accuracy by comparing the current key-frame with the previous frames. This inevitably builds up the computational cost linearly due to the increasing number of processed frames. Although a computer with multi-core processors mitigates such problem to a certain degree, the comparison of the current key-frame to all the earlier frames is not feasible.

Moreover, revisiting the same places only occurs occasionally and a successful loop closure is not always available. Therefore, we adopt a more efficient strategy [5] to select the candidate frames. In order to reduce the number of candidate frames in our system, they are only selected from the set of key-frames. We first detect loop closure in several previous neighbouring key-frames. Subsequently, several key-frames are randomly selected with a preference for much earlier ones to estimate transformation with the current key-frame. When we revisit the same scene and a loop closure is found, more key-frames are further explored in the neighbouring frames of this one to find the best match. Finally the rotation and translation matrix calculated based on the least transformation distance are applied to the global pose optimization process.

3.2.4 Graph Optimization

The edges in the pose graph are generated by transformation estimation between pairs of key-frames. However, they may fail to form a globally consistent trajectory due to estimation errors. In this experiment, we adopt the g2o framework [137] which performs a minimization of non-linear error function. The optimization result can be directly represented as our global trajectory.

In g2o, nodes represent camera poses and edges describe the transformation between camera poses. Assume a local or global loop closure is found, a camera pose k_l can be obtained from both k_i and k_j by

$$\hat{k}_l^i = \hat{R}_{l,i} k_i + \hat{T}_{l,i}, \tag{3.7}$$

$$\hat{k}_l^j = \hat{R}_{l,j} k_j + \hat{T}_{l,j}. \tag{3.8}$$

An error can then be generated by

$$e_{i,j} = \hat{k}_l^i - \hat{k}_l^j. \tag{3.9}$$

The global pose graph is thus optimized by

$$\min_{\hat{R},\hat{T}} \sum_{i,j} ||e_{i,j}||_2^2 \tag{3.10}$$

The error function $e_{i,j}$ is 0 when the estimated transformation matrix is exactly the true value.

In our system, global pose graph optimization is performed when all the key-frames are detected. We have found that graph optimization is of great value when the sensor recaptures the same scene after traveling for a long distance, since the non-linear error function substantially reduces the accumulated noise.

3.2.5 Point Cloud Generation

In order to view the global geometrical 3D map, we need to generate a point cloud, i.e. a set of data points in 3D space. In this experiment, the point cloud is saved as the PCD (Point Cloud Data) file format which can be used by Point Cloud Library (PCL). Similar to a pixel in an RGB image, a point in a point cloud is called a voxel. We first calculate the 3D coordinates of each point in the depth images which has meaningful value (between 0.1 and 5 metres). The voxels in the current point cloud are thus acquired. We then get the colours of the voxels based on their associated pixels in the corresponding RGB images and assign them to the voxels. In this way, a RGB image can be mapped to its corresponding depth image and the point cloud of a key-frame is obtained. We then use the estimated transformation matrix to project the current point cloud into a common coordinate frame and add it to the global point cloud. Finally, the generated global point cloud is down-sampled for better presentation.

3.3 Semantic Information Extraction

We first take some pictures of objects in the environment, associate them with natural language and store them in a database. After extracting SURF descriptors, we take the descriptors of the images in the database and match them with all key-frames. The number of inliers and the Euclidean distance are both used for object recognition, which is similar to loop closure detection.

However, this strategy is not stable in large environments. Furthermore, SURF is good at handling images with blurring and rotation, but not good at handling viewpoint change and illumination change. In Figure 3.2, two key-frames are marked as red dots and their neighbouring key-frames are marked as green dots. A laptop in key-frame A and an error in key-frame B are both labeled as a laptop. To solve this, we simply verify our recognition result in the neighbouring key-frames by

$$\xi_m = \frac{N_c}{N_n} \tag{3.11}$$

where N_c and N_n are the numbers of good matches in the candidate key-frame and the neighbouring key-frames, respectively. A threshold for ξ_m is then set to eliminate recognition errors (in our case the threshold is 1.25). For Key-frame B, ξ_m is larger than 1.25, thus is an error label. The same strategy is also applied to loop closure detection since false positive detection has severe impact on pose graph optimization.

Once an object is successfully detected, the current key-frame is labeled with the object name. After pose graph optimization, the coordinates of the labeled key-frames are recalculated, thus the global coordinates of the recognized objects are obtained.

3.4 Experimental Results

In this section, the 3D mapping and object recognition subsystems are both evaluated. For 3D mapping, a test in our lab is first presented. A Vicon motion tracking system is used to provide the ground truth of our sensor movement. We then test our mapping algorithm on the TUM benchmark [138] and compare our result with the RGB-D SLAM algorithm presented in [5]. Subsequently, we test our system in a large-scale home environment to extract semantic information. We employ a Microsoft Kinect for Xbox One to gather data in our lab and an Asus Xtion PRO LIVE in the home environment. An Intel Core i7-3632QM with 2.2GHz CPU is used for all the experiments. No graphics card is used.

3.4.1 3D Mapping

The mapping result in our lab is shown in Figure 3.3. The lab is approximately circular. The entrance is at the top of the graph. We adopt the benchmark tool provided in [138] to produce

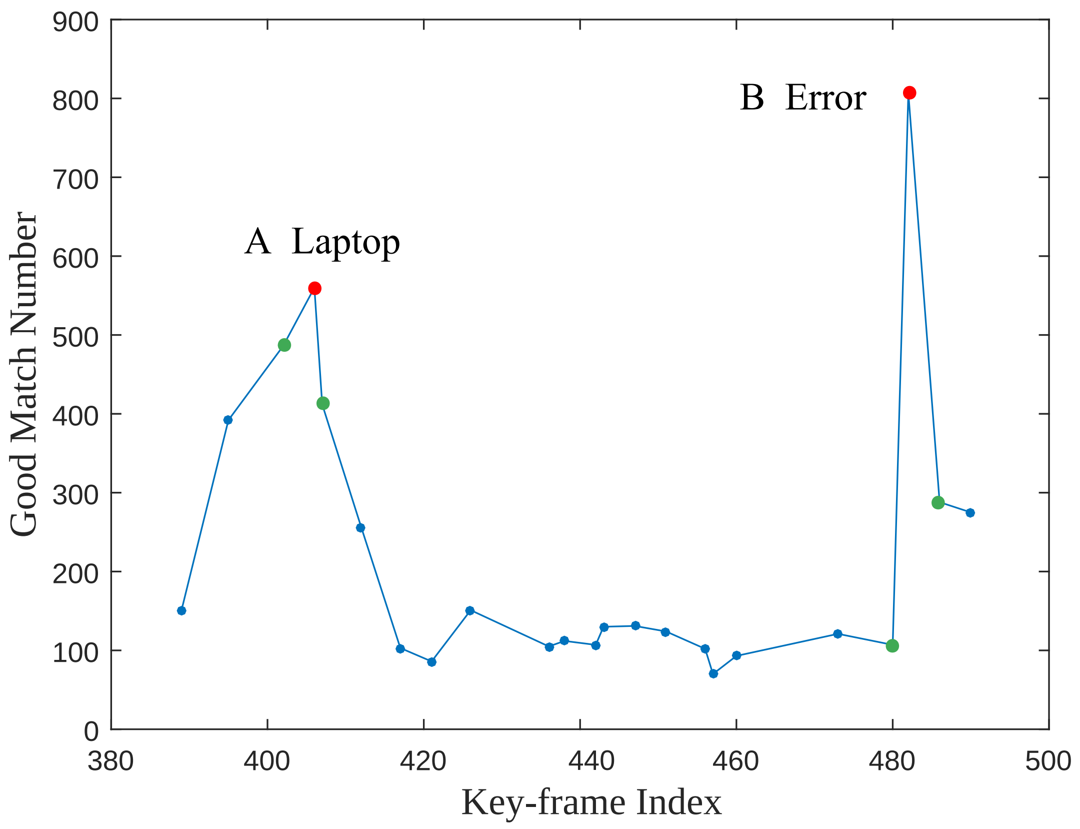
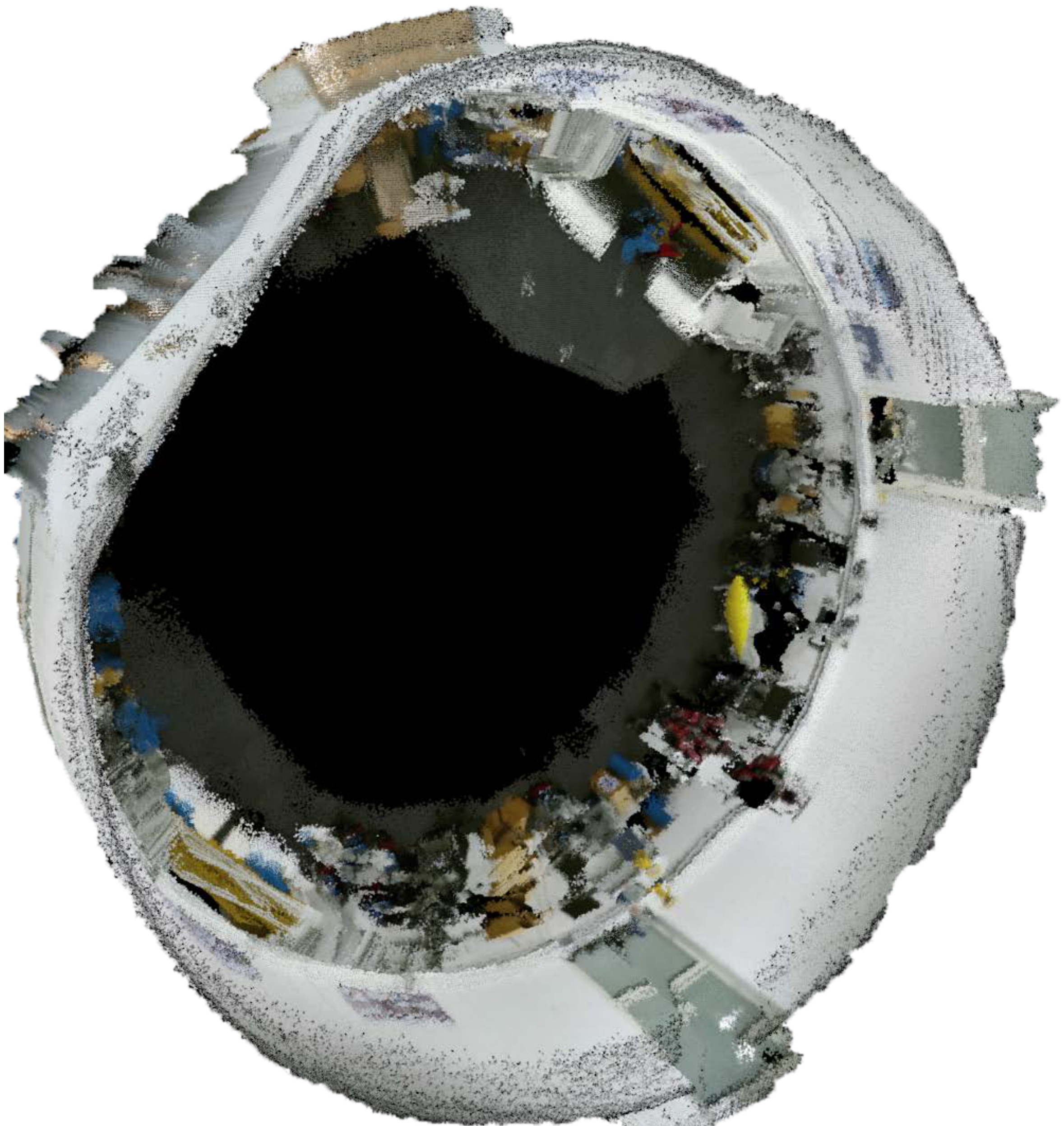


Figure 3.2: Number of good matches.

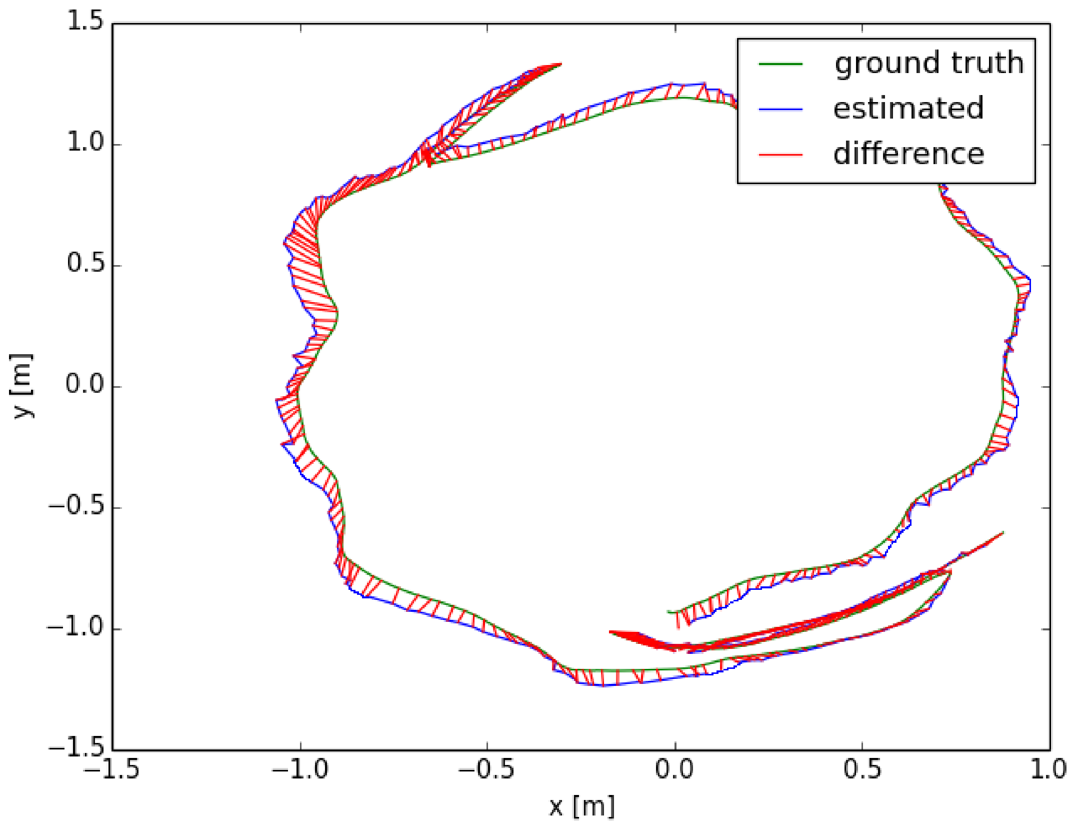
a global trajectory. The green line is the ground truth, the blue line is the estimated pose graph and the red lines show the differences. The top part where the lines intersect indicates when global loop closure is detected. As we can see from the left part of the graph, the differences between the estimated trajectory and the ground truth are relatively greater. This is because the camera was moving too fast, thus the images captured are very blurry.

We have also tested our mapping algorithm on a benchmark dataset. To evaluate our mapping performance, we adopt the absolute trajectory error (ATE) which directly calculates the deviations between pairs of estimated poses and ground truth poses. Both poses are preprocessed and associated using timestamps. In Table 3.1, root-mean-square, mean, median and maximum of ATE are listed. Figure 3.4 presents the 3D maps obtained and the trajectory deviations. We can see that our algorithm can track almost all of the frames and estimate a relatively accurate trajectory.

Figure 3.4h shows the best result. The scene was captured by moving a camera around a table. The proposed algorithm managed to track almost all camera motions smoothly. Thanks to the global loop closure detected, the whole trajectory can be optimized. The algorithm also performed well in Figure 3.4c and 3.4e, as can be seen from Figure 3.4d and 3.4f, respectively. Both scenes were captured by moving a camera along the horizontal and vertical directions in front of tables. Thus, both local and global loop closures can be easily detected. From Figure



(a) 3D map.



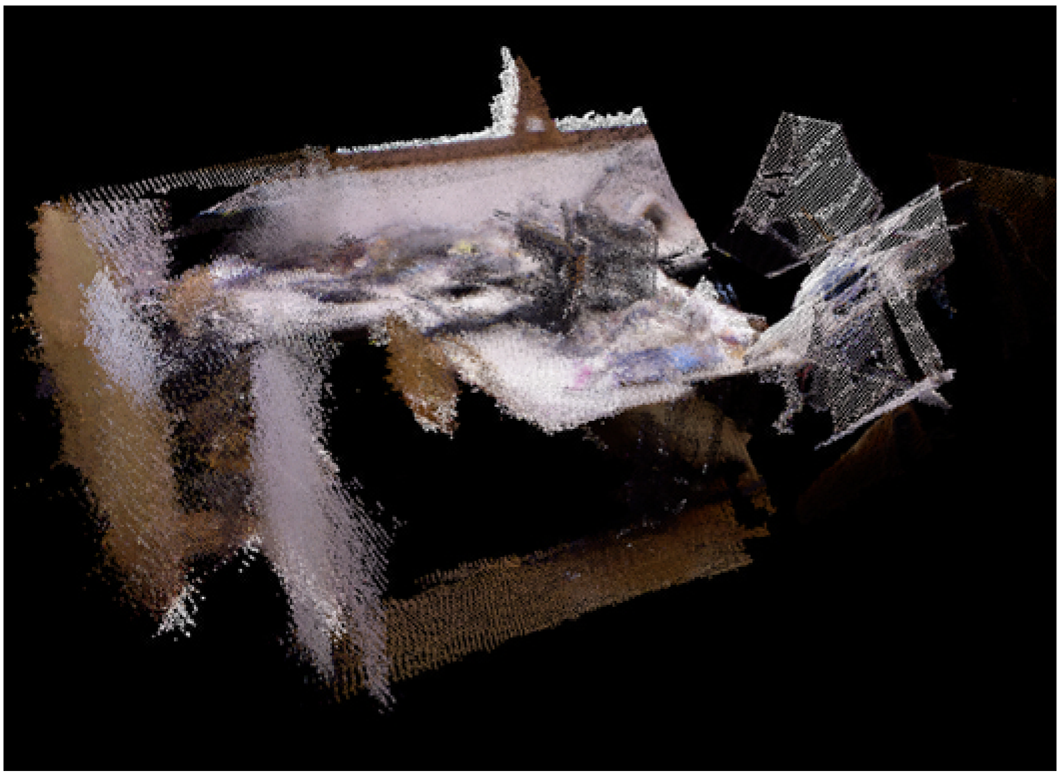
(b) Differences between estimated trajectory and ground truth.

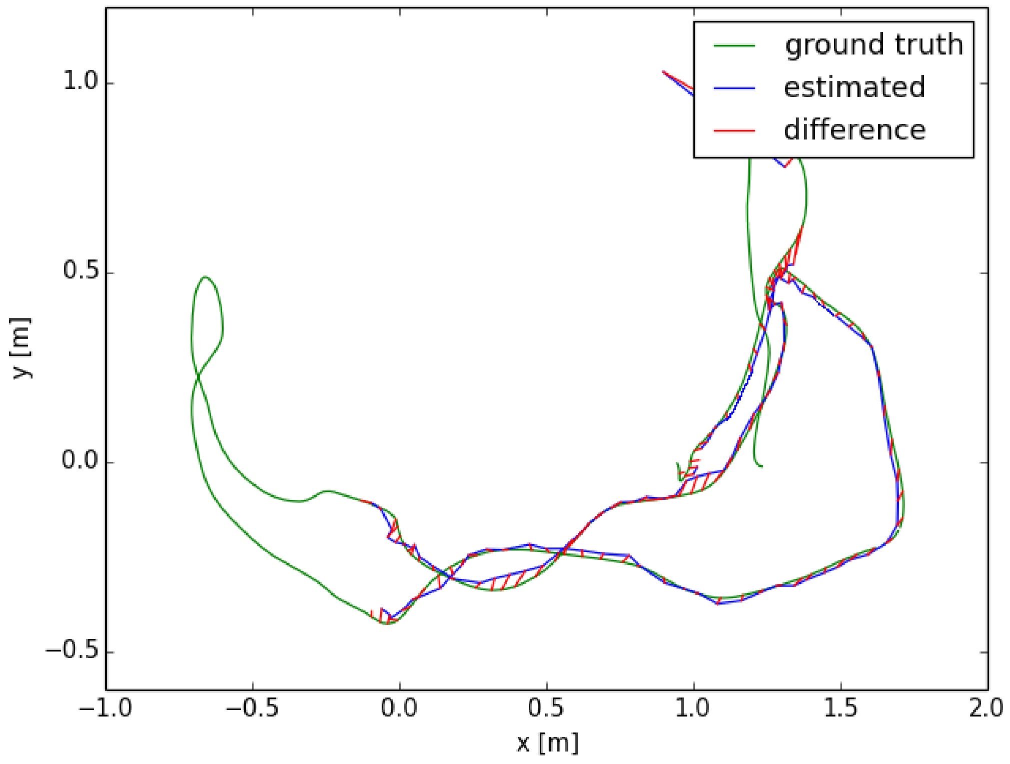
Figure 3.3: 3D mapping test in our lab.

Table 3.1: Performance analysis based on the benchmark and our dataset.

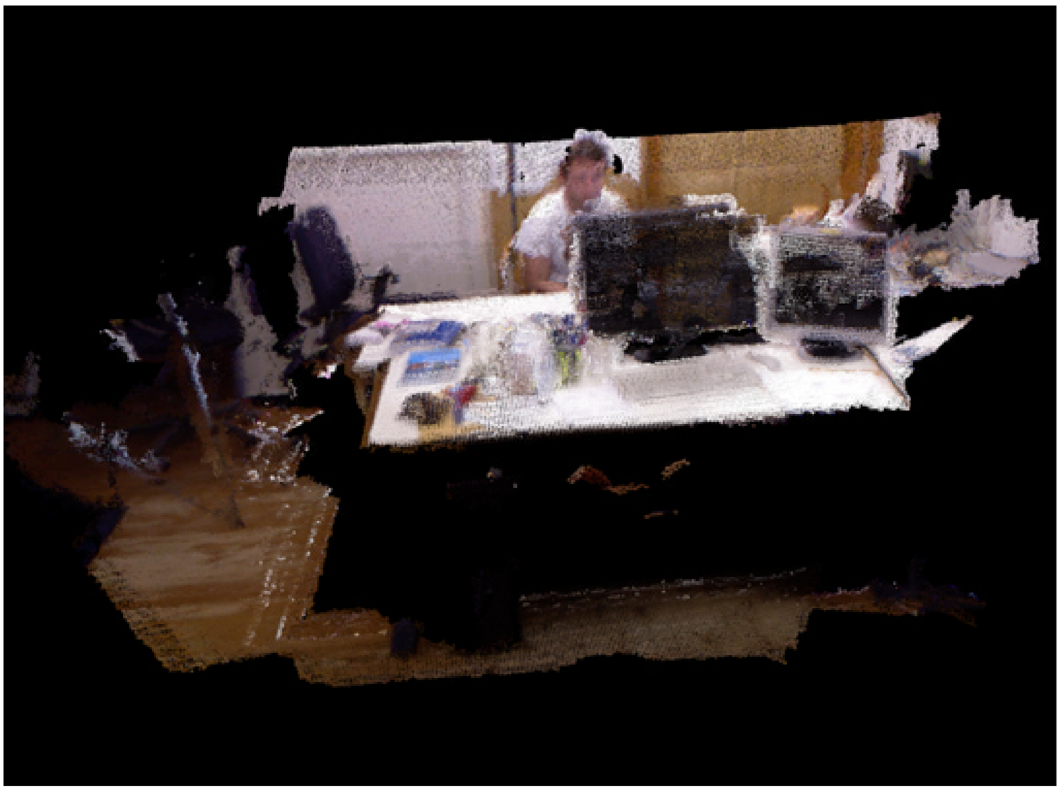
Dataset	Frames	Key-frames	RMSE of ATE (m)	Mean of ATE (m)	Median of ATE (m)	Maximum of ATE (m)
fr1/desk	573	119	0.064	0.034	0.026	0.526
fr1/xyz	792	153	0.013	0.011	0.010	0.051
fr2/xyz	3615	139	0.006	0.005	0.004	0.019
fr3/long	2488	447	0.028	0.027	0.026	0.067
our lab	1528	415	0.055	0.050	0.049	0.133

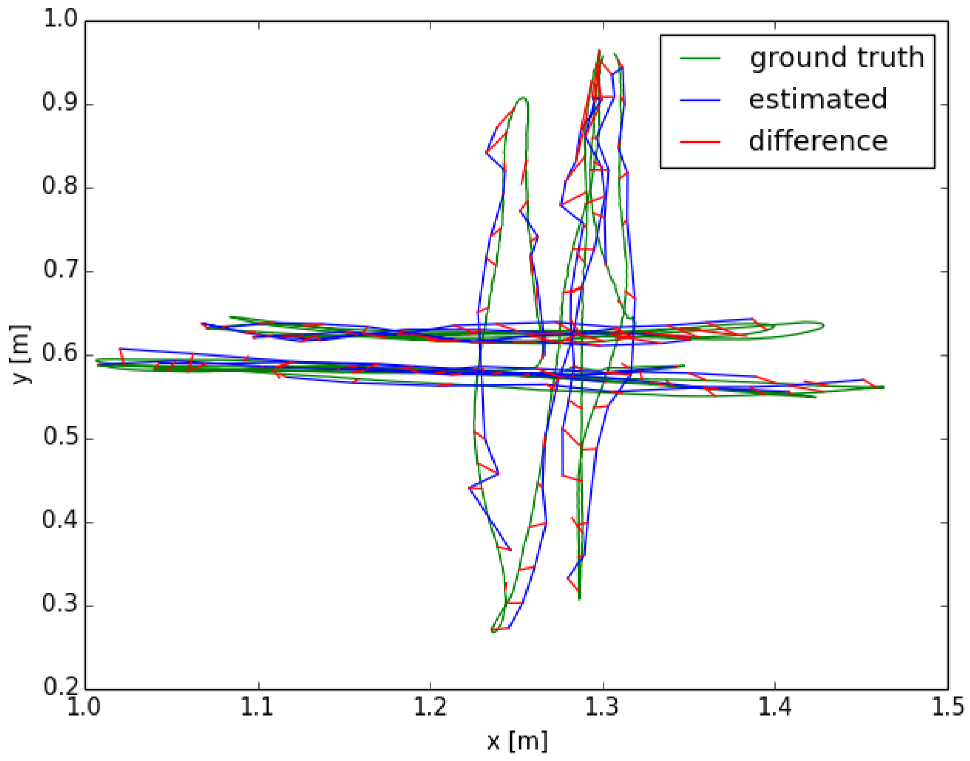
3.4d and 3.4f, we can tell the algorithm omitted some frames during camera pose estimation and at the same time maintained the accuracy. This is because we set a new frame as a key-frame only when the norm of the transformation matrix exceeds a threshold, as detailed in Algorithm 3.1. However, we still failed to track part of the sensor motion in Figure 3.4a, i.e., the left part of the fr1/desk scene. One reason is because the camera was facing to a plain wall when it was capturing the scene, thus there were not enough features extracted from the RGB


(a) fr1/desk map.



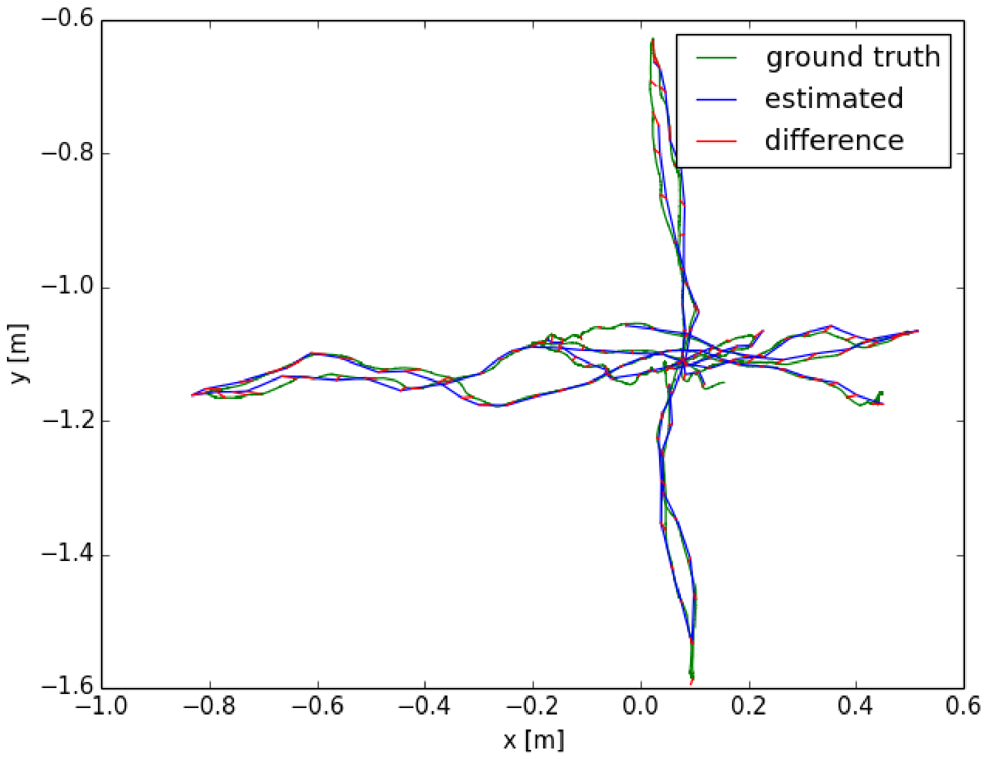
(b) fr1/desk trajectory.


(c) fr1/xyz map.

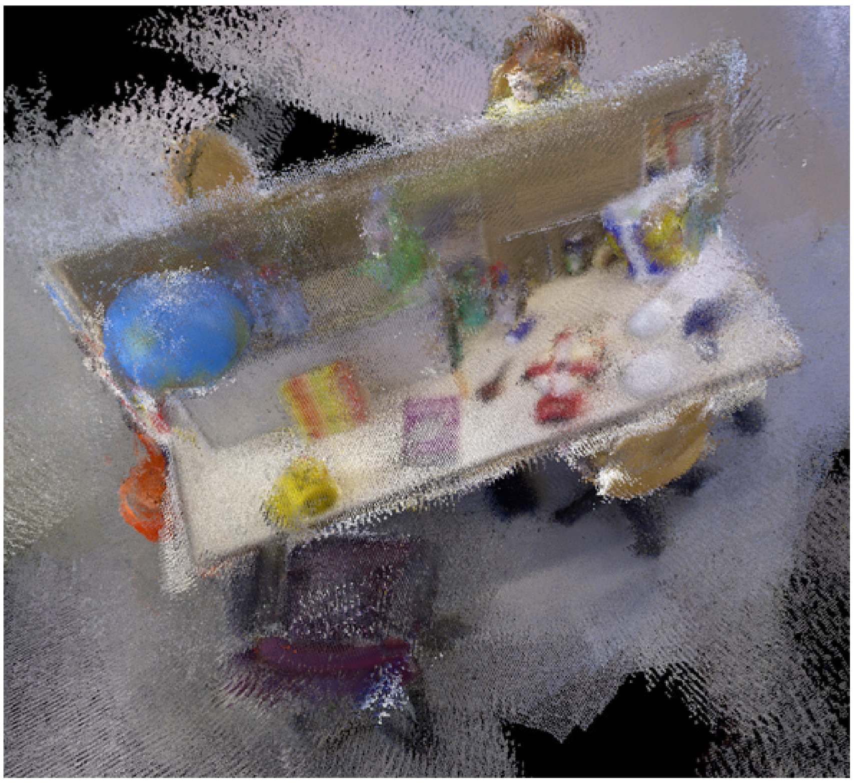


(d) fr1/xyz trajectory.

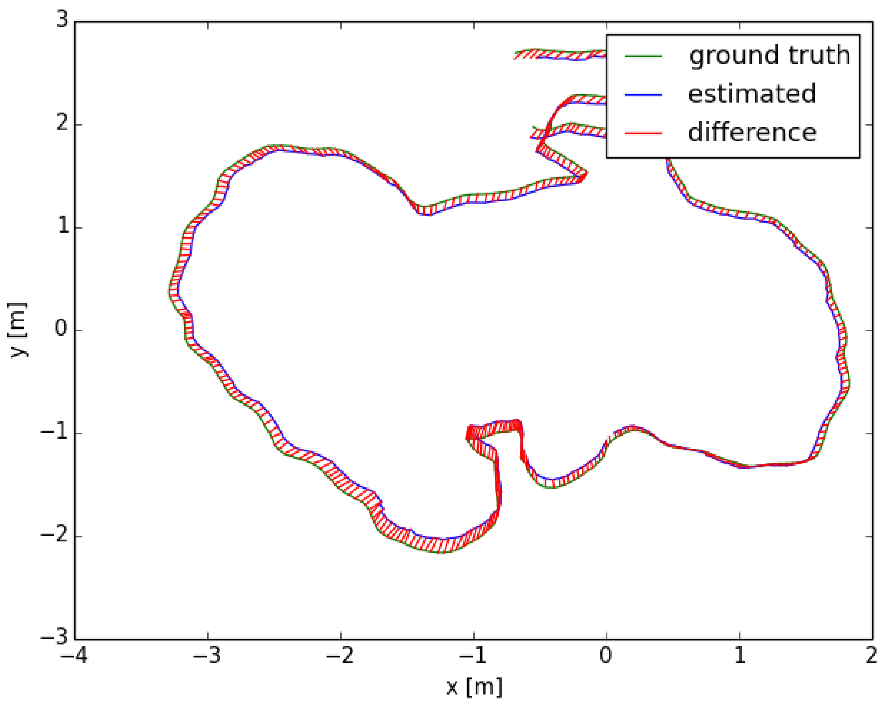
(e) fr2/xyz map.



(f) fr2/xyz trajectory.



(g) fr3/long map.



(h) fr3/long trajectory.

Figure 3.4: 3D mapping test on the benchmark dataset.

Table 3.2: Performance comparison between our algorithm and the one presented in [5] based on RMSE of ATE values. (unit: meter)

Dataset	Our Algorithm	Algorithm in [5]
fr1/desk	0.064	0.026
fr1/xyz	0.013	0.014
fr2/xyz	0.006	0.008
fr3/long	0.028	0.032

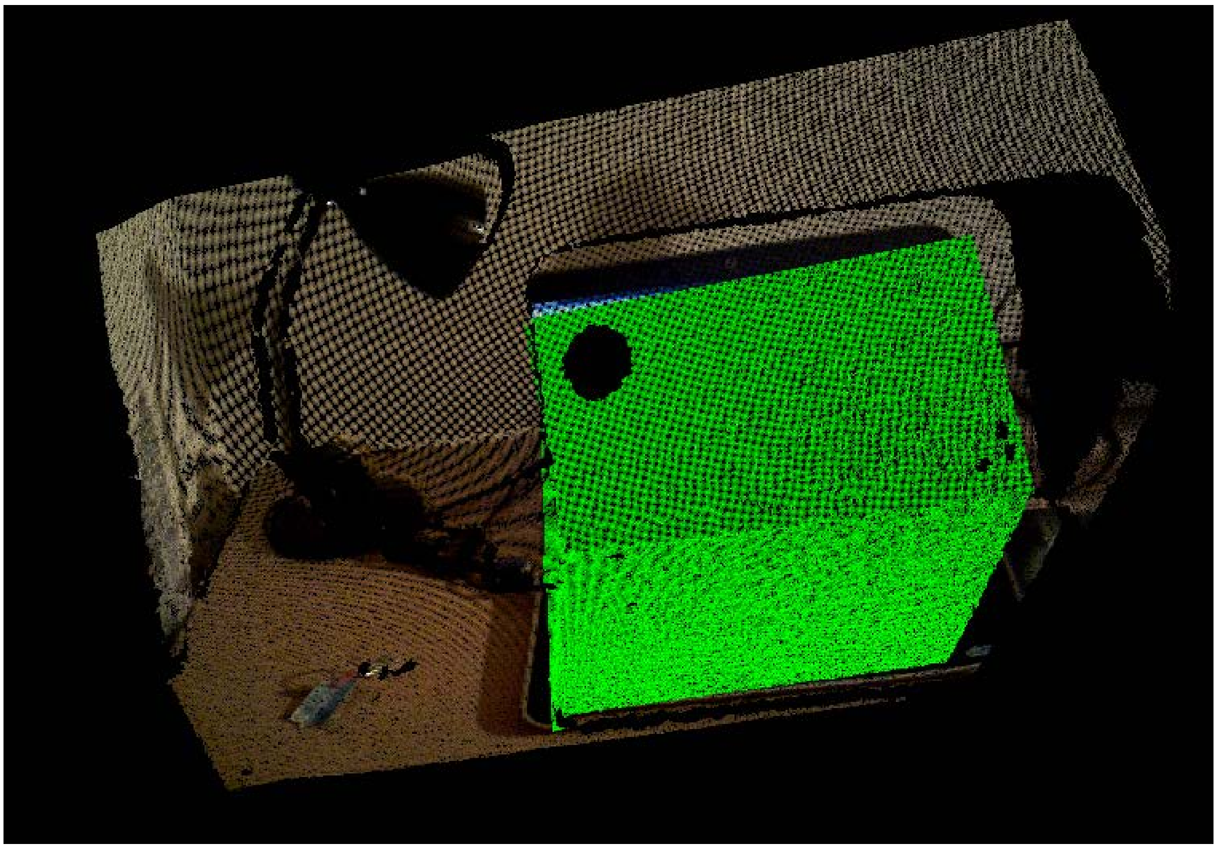


Figure 3.5: A plain colour box showing the recognized laptop in the 3D map.

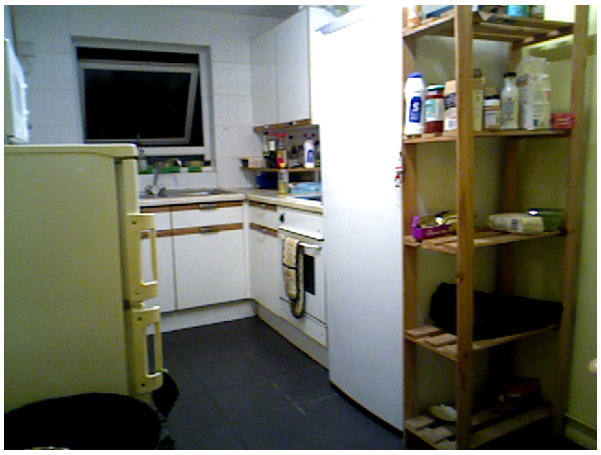
images. Another reason is that the rotation speed of the sensor was too high and the scenes in neighbouring frames suffered from a substantial change. Thus the number of good matches is smaller than the threshold which is used to launch the motion estimation process.

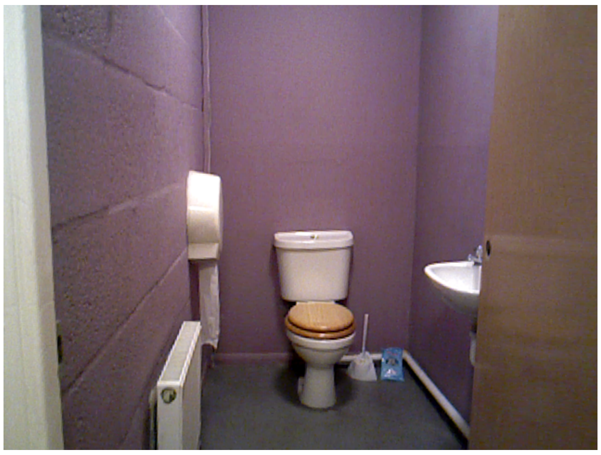
Another mapping algorithm in [5] is used to compare with our results (see Table 3.2). As can be seen from the table, our mapping subsystem has comparable performance as the one in [5]. However, we still need to improve the robustness of our algorithm, especially when the sensor is rotating at a high speed.

3.4.2 Semantic Information Extraction

We carried out the semantic information extraction experiment in a student accommodation including a bedroom, a kitchen and a toilet, as shown in Figure 3.6. In this experiment, 7 objects are added to the database: bed, laptop, toilet seat, stove, kitchen cupboard, kitchen sink and vacuum cleaner. We rely on the handles on the cupboard to recognize it and the tap to recognize the sink. A handheld Xtion sensor and a laptop were used to map the environment.

(a) Bedroom.


(b) Kitchen.


(c) Toilet.

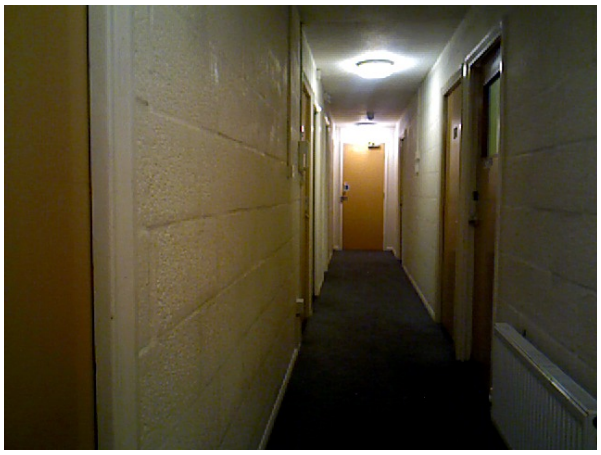

(d) Corridor.

Figure 3.6: Student accommodation.

Since only the centre point of the object features is stored, we place a plain colour box around the centre point to verify the object location, as shown in Figure 3.5. Each object in the database has its own colour. The environment 3D map is shown in Figure 3.7. The red line represents the trajectory of the camera. We started moving the camera from the bedroom, went into the kitchen, walked along the corridor and finally returned to the bedroom. Global loop

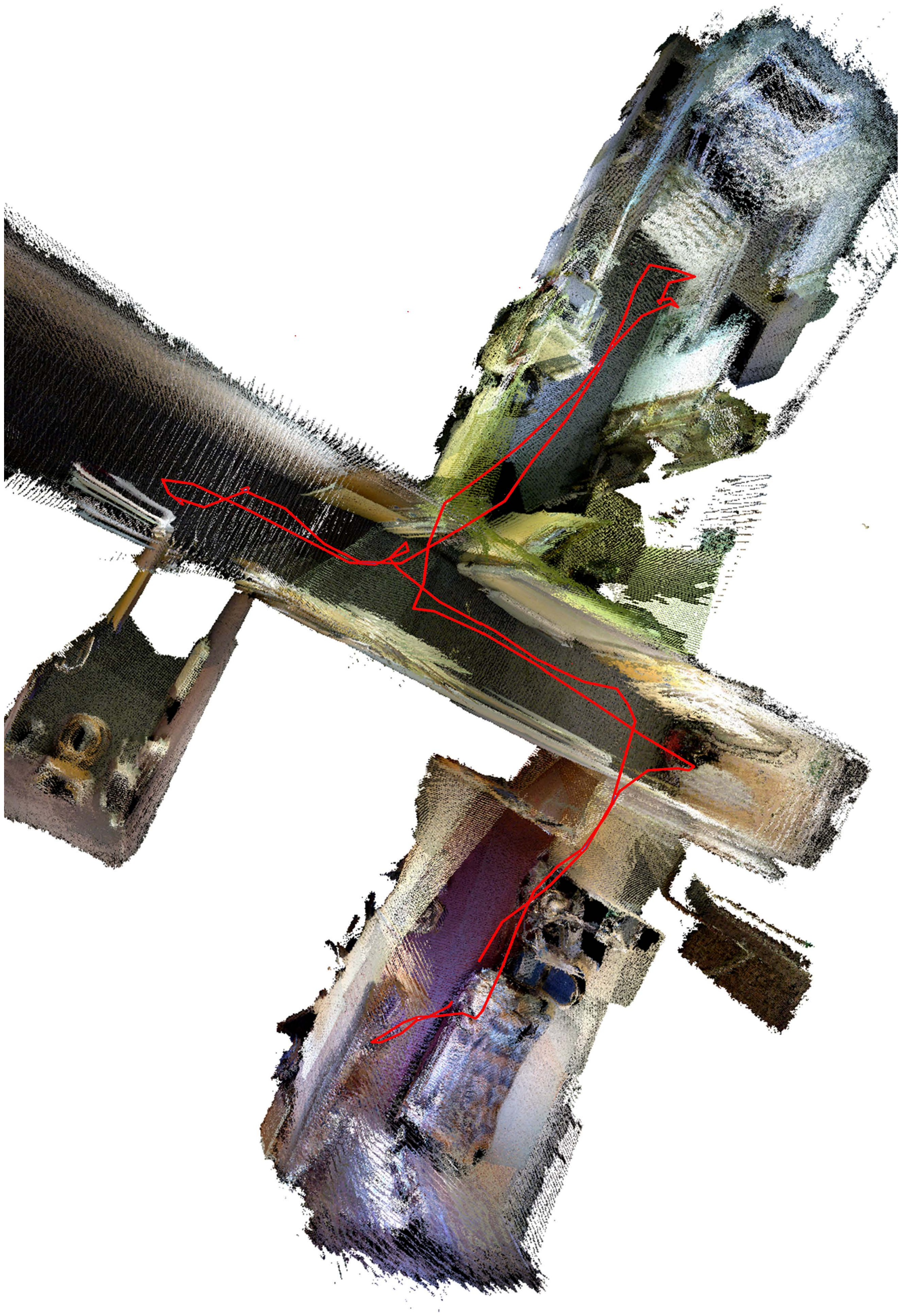


Figure 3.7: 3D map of the student accommodation with red line representing the trajectory of the camera.

closures mainly exist in the corridor.

3.5 Summary

In this chapter, a semantic mapping method was presented to help blind people navigate at home. The system consists of a 3D camera and a laptop. The mapping and semantic information extraction methods were detailed. A novel approach to eliminating errors for loop closure detection and semantic information extraction was also introduced. The pose estimation accuracy was tested and compared based on a benchmark dataset. Finally, the performance of semantic information extraction was verified in a home environment.

In the next section, we continue developing efficient and reliable localization algorithms for semantic mapping. Moreover, deep learning technique will be used in the proposed system to achieve accurate visual localization result.

Chapter 4

Using Unsupervised Deep Learning Technique for Monocular Visual Odometry

This chapter presents a novel monocular visual odometry system based on an unsupervised Recurrent Convolutional Neural Network. In recent years, deep learning based visual odometry systems have already shown promising results compared to traditional feature matching based methods. However, ground truth poses are required for training, which are not always available. Moreover, additional knowledge has to be provided during reconstruction in order to obtain absolute scale from monocular images. To address these issues, we propose a novel visual odometry system using an unsupervised end-to-end framework. Our first contribution is the unsupervised training framework. No camera ground truth poses are required for training. They are only deployed for system performance evaluation. The second contribution is absolute scale recovering without pose post-processing. To inject scale, depth information of scenes is used alongside monocular images to train the network. Poses are inferred only from monocular images, thus making the proposed visual odometry system a monocular one. Experiments have been conducted and the results have shown that the proposed method performs better than other monocular visual odometry systems.

4.1 Introduction

Visual odometry (VO) has drawn enormous attentions from both robotics and computer vision communities during the last decades. It studies how a robot can estimate its movement relative to a rigid scene through a camera (monocular, stereo or omnidirectional) attached to it [139]. Traditional VO systems consist of image correction, feature extraction and representation, feature matching, transformation estimation and pose graph optimization. They have shown some outstanding performance through careful design and adjustment step by step, which are however very costly [22]. The technique has been widely applied to augmented reality (AR), mobile robots, wearable devices, etc.

Deep learning based VO systems developed in recent years [109, 113, 111, 140] have already shown promising performance in terms of both translation and rotation estimation accuracy. Ground truth poses of each input frame need be acquired beforehand and fed into these networks for training. However, ground truth poses are difficult and expensive to obtain. In some systems, ground truth poses are even inferred and obtained by labeling collected images with traditional VO or SLAM algorithms, which is an ill-posed problem.

This chapter proposes an unsupervised training framework which does not require the ground truth poses of a camera in any form for training. Instead, the ground truth poses of the camera are only used for performance evaluation. Therefore, such unsupervised training eliminates the need of the labour-intensive image labeling process. In addition, the performance of our system can be easily improved by further training with larger unlabeled dataset. Figure 4.1 gives an overview of our proposed VO system. The upper half of the figure shows the training pipeline, whereas the lower half shows the testing pipeline. The black lines represent the inputs of the proposed Recurrent Convolutional Neural Network, the blue lines represent the outputs and the red line represents back-propagation. The training dataset includes a pair of monocular and depth images. Transformation matrices generated by the network are used to calculate losses. Parameters in the network are then optimized by minimizing these losses. We use consecutive monocular images for testing. The network directly yields poses on an absolute scale.

Monocular VO is one of the most popular VO categories depending on the camera setup. However, the absolute scale can not be obtained based solely on monocular images. Either external information or prior knowledge (ground truth pose) is required at some stage during

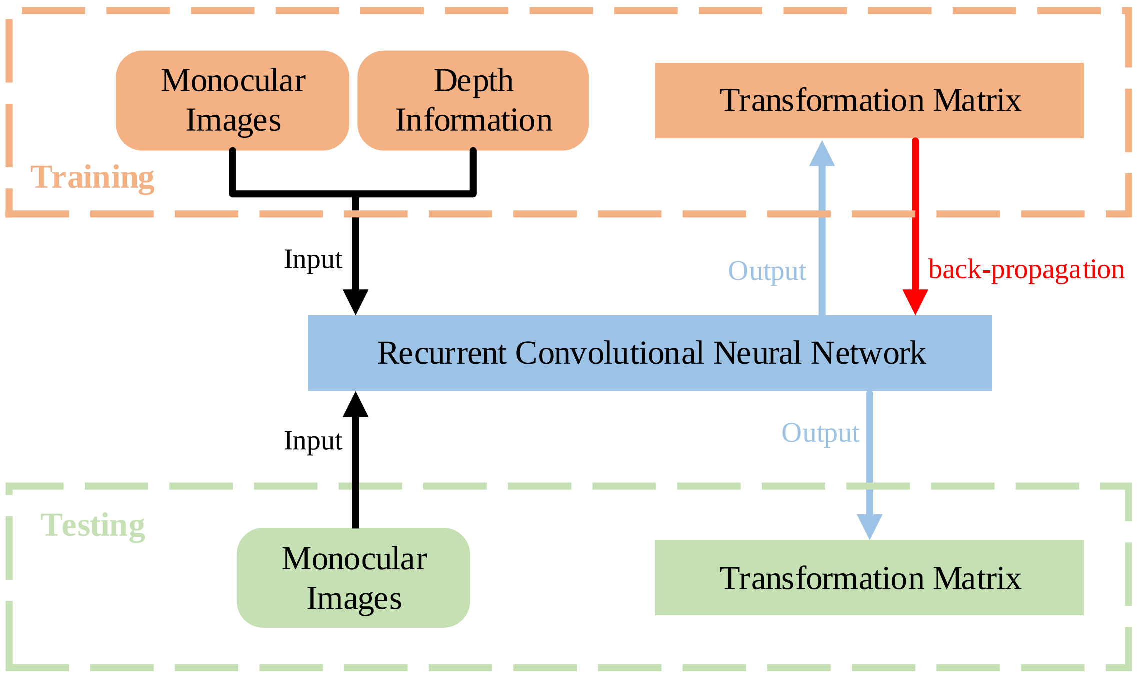


Figure 4.1: Overview of the proposed visual odometry system.

reconstruction or/and training. In robotics, one typical way of obtaining scale during reconstruction is by combining a monocular camera with other sensors such as Inertial Measurement Unit (IMU) and optical encoder. Another solution is by providing depth information of a scene in some way. This can be achieved via employing RGB-D sensors (Microsoft Kinect, Asus Xtion Pro, etc.) [141, 142, 143], stereo cameras [144, 145] or 3D LiDARs [146, 147]. This latter one has been widely deployed in self-driving cars and smart phones.

In this chapter, we feed monocular images and depth information obtained from 3D LiDARs into the training pipeline to inject absolute scale and only use monocular images during testing. We focus on the problem of continuously localizing a monocular camera on an absolute scale for the purpose of locating people or robots. The RCNN is trained based on an unsupervised end-to-end manner. Experiments have been carried out on KITTI [148] odometry dataset and results have shown that our VO system can be compared to other state-of-the-art monocular VO systems in terms of both translation and rotation accuracy even without scale post-processing.

In general, visual odometry tackles the problem of recovering the position and orientation of an agent or a robot in 3D world from associated images. Based on the type of camera employed, VO systems can be divided into several categories, namely monocular VO [149], stereo VO [150] and omnidirectional VO [151]. Additional sensors are sometimes incorporated to boost the performance, such as depth sensors [71] (LiDAR or RGB-D camera) and IMU [152].

Most traditional VO systems are feature based. More specifically, certain image features are extracted and represented by descriptors first. They are then matched across a sequence of images and used to calculate transformation matrices between frames. The performance of these systems depends heavily on the image features deployed. Speeded Up Robust Features (SURF) and Scale Invariant Feature Transform (SIFT) features were used by Kitt *et al.* [153] and Barfoot [154] in their stereo VO systems respectively. Mur-Artal *et al.* modified Oriented FAST and rotated BRIEF (ORB) feature and proposed one of the state-of-the-art SLAM systems [155, 156].

ORB-SLAM is superbly fine-tuned and can be operated in real-time without GPUs. Such systems are built on the idea of parallel tracking and mapping (PTAM) [157]. They are computationally efficient since a whole image is represented by a sparse set of feature observations and only the features are involved in calculation. An alternative to feature based method was brought up by Newcombe *et al.* [134, 158], namely dense tracking and mapping (DTAM), which can be viewed as a direct method. DTAM relies on pixel intensity and minimizes an error directly in sensor space. Therefore, feature extraction and matching are not required.

However, due to the high computational demand of processing every pixel in an image, GPUs inevitably need to be employed to make the system run in real-time. Engel *et al.* proposed a hybrid semi-dense system, namely LSD-SLAM, which is operated in real-time with only a CPU while maintaining the accuracy and robustness of dense approaches [159, 160]. LSD-SLAM first builds up an inverse depth map of an image for camera motion estimation. The inverse depth map is semi-dense, which is estimated from the image regions with severe gradient changes rather than a whole image. In this way, the texture of the image is preserved and the computational complexity can be significantly reduced. These systems usually need to be carefully designed and fine-tuned. In contrast, our method adopts an end-to-end training framework and requires less engineering effort.

In recent years, Convolutional Neural Networks (CNNs) have been widely used in the robotics and computer vision domain and have shown remarkable robustness in challenging environments [107]. This is due to the more descriptive features extracted and the extremely large and diverse data used for training. PoseNet proposed by Kendall *et al.* shows the first implementation on pose estimation [109], which directly generates the six degrees of freedom (6-DoF) of a camera from a single RGB input image. The model GoogLeNet pre-trained on other classification tasks is leveraged for pose regression [110]. The softmax layers that originally output classification results are removed and replaced by a seven-dimensional pose vector. The last fully connected layers are also modified.

CNNs extract more robust features than traditional feature detectors and achieve a high accuracy even under some extreme conditions, such as intense lighting and blurry images. PoseNet can also be easily generalized to other scenes through transfer learning technique. The model on the new task can thus be trained with smaller dataset and shorter time. Li *et al.* incorporated another CNN stream to PoseNet and fed depth images into this stream to enhance the re-localization accuracy [111]. ORB-SLAM is used to label the collected images as ground truth. Recurrent Convolutional Neural Networks [112] were also employed by Wang *et al.* [113] for pose estimation. However, all of these deep learning based methods require ground truth for training, which can be quite expensive and labour-intensive.

Attentions have been recently drawn to the unsupervised field due to the shortcomings of the aforementioned supervised methods. Zhou *et al.* presented an unsupervised deep learning framework for depth and camera motion estimation [147]. An explainability mask is also trained to prevent gradient corruption. Their depth prediction and the pose estimation results were promising. However, this method failed to recovery absolute scale due to the limitation caused by using monocular images only. A scale factor needs to be calculated from ground truth each time when a pose is estimated and the value of the scale factor is non-constant.

The rest of Chapter 4 is organized as follows. The proposed network architecture and the methodologies are detailed in Section 4.2. Training and experimental results are subsequently presented and evaluated in Section 4.3. Finally, a brief conclusion is given in the last section.

4.2 The Proposed Approach

In this section, we discuss the proposed VO system in detail. The network architecture is given first. The loss functions used to penalize the system output are subsequently introduced. Finally, the implementations of the network and loss functions are presented.

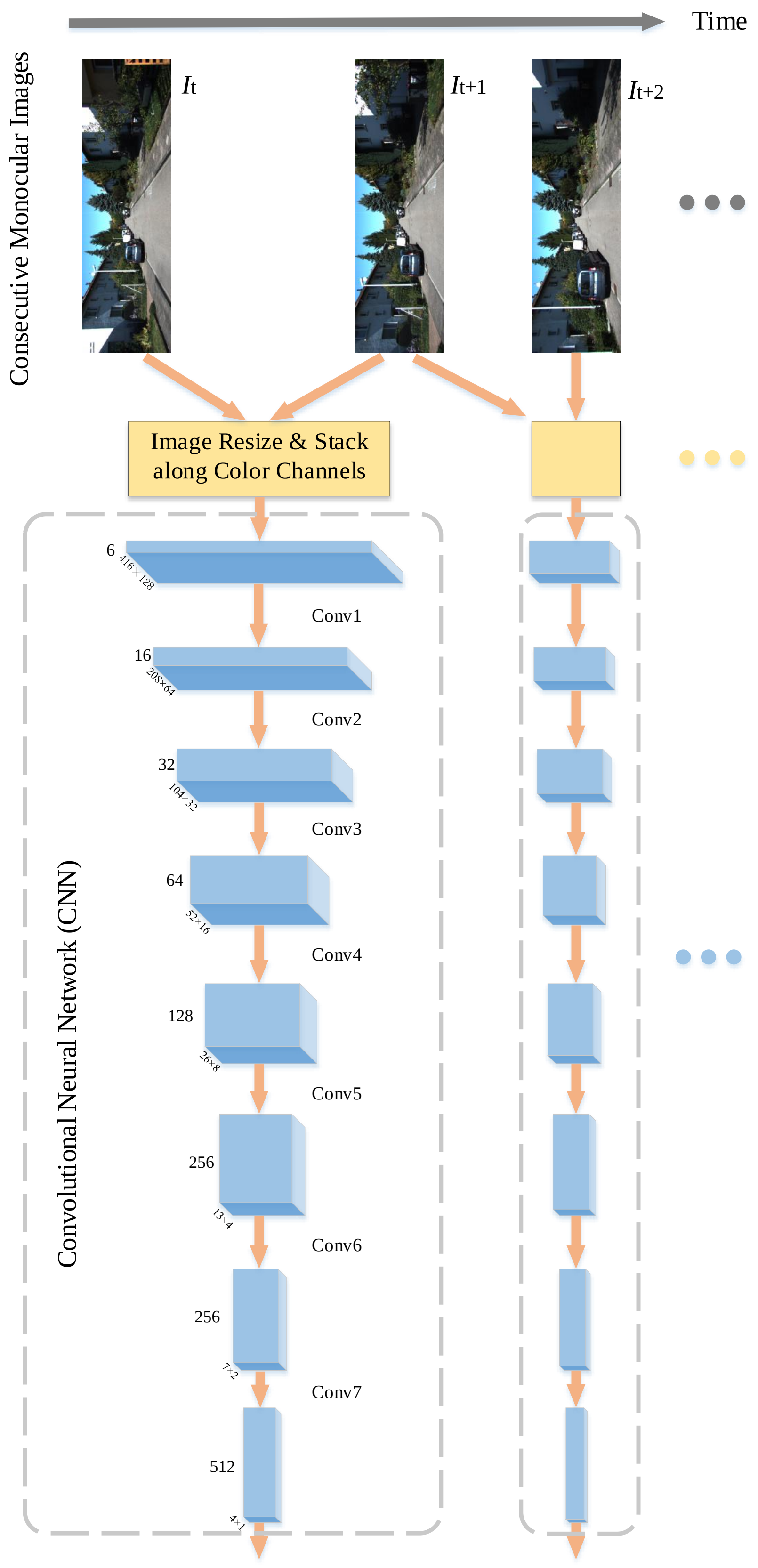


Figure 4.2: Architecture of the Convolutional Neural Network with input images.

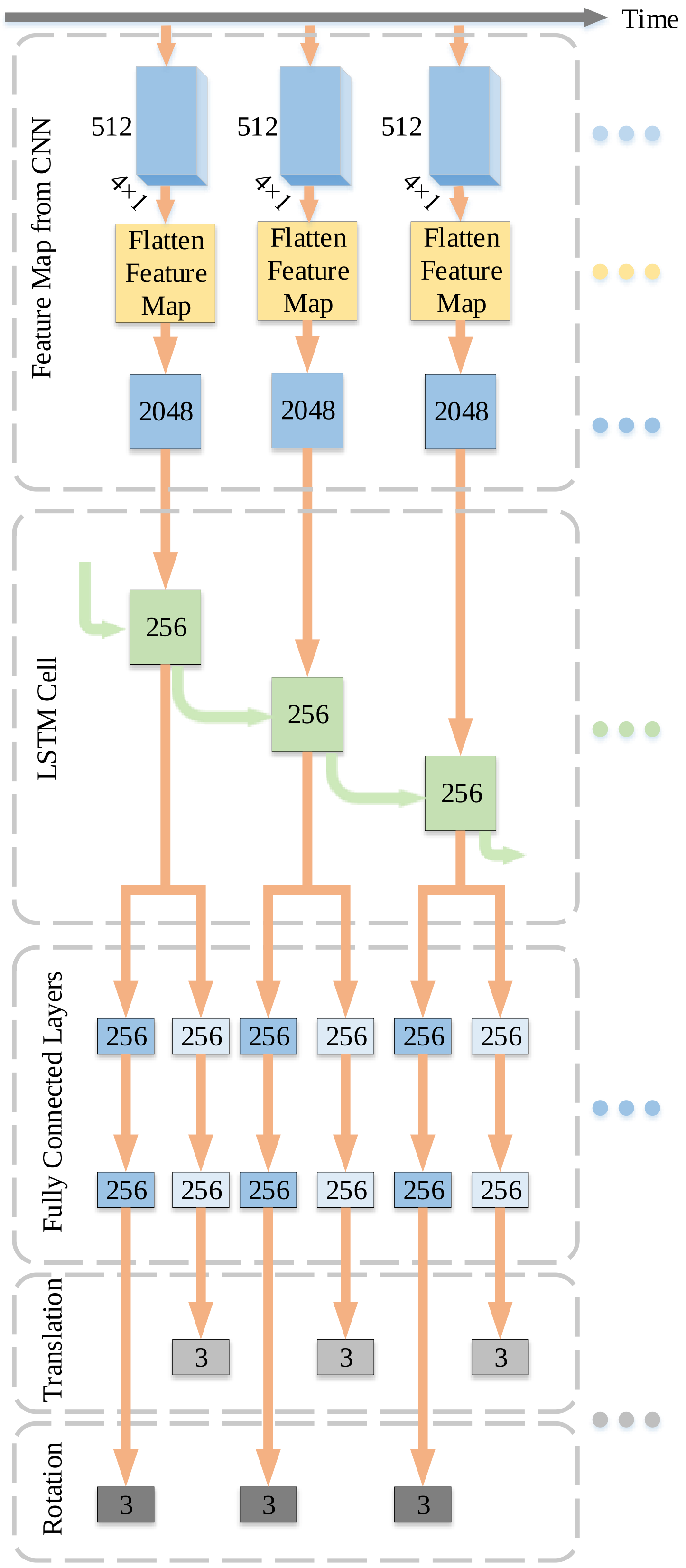


Figure 4.3: Architecture of the Recurrent Neural Network with output poses.

4.2.1 System Architecture

Visual Odometry describes the movement of an agent over time. The global pose graph is obtained from a sequence of images gradually rather than through a single calculation. Every element of the image sequence is not independent of each other. Therefore, the deep learning network needs to consider the previous computations before it outputs the pose of the current frame. With regard to current neural networks, a CNN, being a feed-forward network, only learns to differentiate patterns across space, while a RNN learns to recognize patterns across time. Leveraging both CNN and RNN networks could meet the requirements of the VO task perfectly. Thus, following the methodology presented by Wang *et al.* [113] and Donahue *et al.* [112], we propose a Recurrent Convolutional Neural Network (RCNN) in this section. The CNN takes two raw RGB images as input and generates a feature map. The feature map is then fed into the RNN which finally generates a transformation matrix between the input images.

Figure 4.2 shows the architecture of the Convolutional Neural Network. The network can be viewed as a feature extractor. We take two consecutive monocular images each time and feed them into the network for training. The images are resized to $416 \times 128 \times 3$ and then stacked along colour channels. *Conv* represents convolutional layers. The blue cubes represent feature maps with shapes under them. Figure 4.3 shows the architecture of the Recurrent Neural Network. The network can be viewed as a pose estimator. The network takes the last feature maps from the CNN and directly outputs translation and rotation matrices. The numbers in blue and gray boxes represent the size of vectors. The number of hidden units in a LSTM cell is set as 256.

It becomes clear that CNNs that are originally trained for a specific task can be modified and reused for other related but different tasks [161, 105] since the generic features learned by a model, especially from lower convolutional layers, are versatile and transferable [107]. Recently, several models have been proposed and shown promising performance such as AlexNet [162], GoogLeNet [110] and ResNet [163]. Based on the CNN originates from Visual Geometry Group neural network (VGG) [164], Table 4.1 lists the specifics of each modified convolutional layer.

Figure 4.2 uses KITTI dataset as an example input. The CNN model can be regarded as an image feature extractor and descriptor. Assume that $I_1, I_2, ..., I_t, ..., I_N$ is a sequence of

Table 4.1: Specifics of the convolutional layers.

Layer	Filter Size	Stride	Padding	Channel Number
Conv1	7×7	2	3	16
Conv2	5×5	2	2	32
Conv3	3×3	2	1	64
Conv4	3×3	2	1	128
Conv5	3×3	2	1	256
Conv6	3×3	2	1	256
Conv7	3×3	2	1	512

monocular images used for training. The CNN takes every two consecutive images as input and yields $N - 1$ feature maps with the size $4 \times 1 \times 512$. The input images are first resized to $416 \times 128 \times 3$, stacked along colour channels and then fed into the network. There are 7 convolutional layers in the CNN. We use stride 2 to regulate the movement of all of the convolutional filters (receptive field or kernel) for pixel-wise operations across image space. The sizes of the filters in the first two convolutional layers are 7×7 and 5×5, respectively. The size drops to 3×3 for the rest layers. The zero-padding decreases along with the kernel size from 3 to 2 and then 1 so that the spatial dimension of the input volume can be preserved.

Each convolutional layer is followed by a Rectified Linear Unit (ReLU) nonlinear activation function. Batch normalization, which is a commonly used technique for improving performance of neural networks is not employed in our CNN. Instead, it results in slow and unstable loss convergence in our experiments. One possible reason is because batch normalization normalizes the input layer by adjusting and scaling the activations. The absolute differences between image pixels or features are ignored and only relative differences are taken into consideration. In this way, batch normalization can reduce the training difficulty for classification tasks since it can retain the structure of an image while highlighting the inconspicuous regions. However, the contrast information of an image needs to be preserved rather than stretched for VO tasks. Thus, batch normalization is not applied in our system.

The feature maps generated from the CNN are reshaped and flattened to $N - 1$ chronological vectors. The RNN takes these vectors as input and learns connections in the sequence of image. However, in practice, it is difficult to train a standard RNN to solve problems that require learning long-term temporal dependencies, since the gradient of the loss function decays exponentially with time until it vanishes or explodes. Thus, we adopt a popular solution by incorporating Long Short-Term Memory (LSTM) units [165] into the RNN.

Compared to standard RNNs, LSTM networks introduce three gates, namely input, forget and output, which allow for a better control over the gradient flow and preservation of long-term temporal dependencies. The key to an LSTM network is updating the cell state through time, which is represented by the green arrows in Figure 4.3. Only one LSTM layer is applied in the RNN and the number of memory units is 256. We follow Kawakami's suggestion [166] and set the biases of the forget gate to 1 to reduce the scale of forgetting at the beginning of training. The projection layer is not used in the LSTM cell, thus the dimension of the output is also 256.

The output vectors from the RNN represents high-level features of the transformation information between two consecutive frames. We then feed them into two fully connected layers to learn nonlinear combinations of these features. The fully connected layers have connections to all activations in the previous layer, thus can realize high-level meaningful reasoning. Unlike other deep learning based methods which output a single vector representing 6-DoF, two parallel streams are introduced in our system to infer translation and rotation independently. This is due to the fact that rotation is highly nonlinear and is always harder to be trained. A traditional solution based on practical experience is by raising the weight of rotation loss. We further extend this idea and use two separate streams to collect different features for estimation. The dimension of the fully connected layers is 256, followed by a Exponential Linear Unit (ELU) activation function. Finally, the translation and rotation (represented by Euler angles) vectors are generated and used for back-propagation.

4.2.2 Loss Function

In this section, we introduce how the loss functions are designed in our system. The loss functions or cost functions describe how far off the pose our RCNN produced is from the expected result. The loss indicates the magnitude of error our model made on its inaccurate prediction. We minimize the loss in order to make the output of the network closer to the truth. In our system, the total loss consists of 2D and 3D spatial losses. The loss is calculated by using transformation matrices generated by our RCNN and pairs of consecutive monocular images and point clouds.

Assume that $I_1, I_2, ..., I_t, I_{t+1}, ..., I_N$ is a sequence of monocular images in chronological order used for training and $D_1, D_2, ..., D_t, D_{t+1}, ..., D_N$ are the associated depth images. I_t and $I_{t+1}, (1 \leq t < t+1 \leq N)$ are two consecutive frames in this image sequence. To compute 2D spatial loss, we first project a point from I_t to I_{t+1} using the transformation matrix and its depth value. A new frame $\hat{I}_t$ can then be reconstructed from the projected point in I_{t+1}. Finally, we compare $\hat{I}_t$ with I_t for loss calculation. In terms of 3D spatial loss, we directly swap a point cloud to its neighbouring frame through the transformation matrix and compare their difference.

2D Spatial Loss

Pairs of consecutive RGB images and point clouds are used to compute 2D spatial loss. We first rescale the voxel values of the point clouds to 0-255 and project the point clouds to single-channel 2D depth images. Thus, each pixel in a calibrated depth image represents the depth value of the corresponding point in the associated monocular image.

Let $p_t(u_t, v_t)$ and d_t denote a point in I_t and its depth value in D_t, respectively. We then try to project p_t to the frame I_{t+1} at time $t+1$. Assume the projected point in I_{t+1} is $\hat{p}_{t+1}(\hat{u}_{t+1}, \hat{v}_{t+1})$. Based on the pinhole camera model, a scene view can be formed by projecting 3D points in the world coordinate system into the image 2D plane using a perspective transformation

$$d_t p_t = K P_t \tag{4.1}$$

or

$$d_t \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = K \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}, \tag{4.2}$$

where K is the camera intrinsic matrix, $P_t(x_t, y_t, z_t)$ is the voxel in the world coordinate system projected from point p_t. Note that in Equation 4.2, $d_t = z_t$.

On the other hand, based on 3D linear transformation theory, we have

$$\hat{P}_{t+1} = \hat{R}_{t->t+1} P_t + \hat{t}_{t->t+1} \tag{4.3}$$

or

$$\hat{P}_{t+1} = \hat{R}_{t->t+1} d_t K^{-1} p_t + \hat{t}_{t->t+1}, \tag{4.4}$$

where $\hat{P}_{t+1}$ is the voxel in the world coordinate system projected from $\hat{p}_{t+1}$. $\hat{R}_{t->t+1}$ (converted from Euler angles) and $\hat{t}_{t->t+1}$ are the rotation matrix and translation vector generated by the RCNN, respectively. The size of rotation matrix $\hat{R}_{t->t+1}$ is 3×3, whereas the size of translation vector $\hat{t}_{t->t+1}$ is 3×1. We can then project $\hat{P}_{t+1}(\hat{x}_{t+1}, \hat{y}_{t+1}, \hat{z}_{t+1})$ to the image 2D plane through

$$\hat{d}_{t+1} \begin{bmatrix} \hat{u}_{t+1} \\ \hat{v}_{t+1} \\ 1 \end{bmatrix} = K \begin{bmatrix} \hat{x}_{t+1} \\ \hat{y}_{t+1} \\ \hat{z}_{t+1} \end{bmatrix}, \tag{4.5}$$

where $\hat{d}_{t+1}$ is the depth value of $\hat{p}_{t+1}(\hat{u}_{t+1}, \hat{v}_{t+1})$ and $\hat{d}_{t+1} = \hat{z}_{t+1}$. In this way, we can derive $\hat{p}_{t+1}$ from p_t by

$$\hat{p}_{t+1} = \frac{1}{\hat{z}_{t+1}} K(\hat{R}_{t->t+1} d_t K^{-1} p_t + \hat{t}_{t->t+1}). \tag{4.6}$$

We then use the framework proposed by Jaderberg *et al.* [167] to reconstruct I_t. More specifically, the value of p_t in the reconstructed image $\hat{I}_t$ is generated by the top left, top right, bottom left and bottom right neighbours of $\hat{p}_{t+1}$ in I_{t+1}. Similarly, we can reconstruct image I_{t+1} by

$$\hat{p}_t = \frac{1}{\hat{z}_t} K(\hat{R}_{t+1->t} d_{t+1} K^{-1} p_{t+1} + \hat{t}_{t+1->t}), \tag{4.7}$$

where p_{t+1} is a point in I_{t+1}, $\hat{p}_t$ is the projected point in I_t, d_{t+1} is the depth value of p_{t+1}, $\hat{P}_t(\hat{x}_t, \hat{y}_t, \hat{z}_t)$ is the voxel in the world coordinate system projected from $\hat{p}_t$, $\hat{R}_{t+1->t} = \hat{R}^{-1}_{t->t+1}$, $\hat{t}_{t+1->t} = -\hat{R}^{-1}_{t->t+1} \hat{t}_{t+1->t}$.

Finally, the 2D spatial loss can be represented by

$$L_{2D} = \sum_{t=1}^{N-1} \left(\overline{|I_t - \hat{I}_t|} + \overline{|I_{t+1} - \hat{I}_{t+1}|} \right). \tag{4.8}$$

3D Spatial Loss

3D spatial loss is computed by using point clouds and transformation matrices generated from the RCNN. Assume C_t and C_{t+1}, $(1 \le t < t+1 \le N)$ are two consecutive point clouds which are inverse-projected from D_t and D_{t+1} to the world coordinate system. Let c_t denote a point in C_t, we then project this point to C_{t+1} through transformation matrix. Based on 3D linear transformation theory, the projected point can be derived by

$$\hat{c}_{t+1} = \hat{R}_{t->t+1} c_t + \hat{t}_{t->t+1}. \tag{4.9}$$

The reconstructed point cloud $\hat{C}_{t+1}$ can thus be obtained. We can also reconstruct $\hat{C}_t$ from C_{t+1} by

$$\hat{c}_t = \hat{R}_{t+1->t} c_{t+1} + \hat{t}_{t+1->t}, \tag{4.10}$$

where c_{t+1} is a point in C_{t+1} and $\hat{c}_t$ is the projected point in $\hat{C}_t$.

Finally, we employ a strategy similar to Iterative Closest Point (ICP) algorithm proposed by Chen *et al.* [168] for 3D spatial loss calculation,

$$L_{3D} = \sum_{t=1}^{N-1} \left(\overline{|C_t - \hat{C}_t|} + \overline{|C_{t+1} - \hat{C}_{t+1}|} \right). \tag{4.11}$$

The total loss can thus be acquired by

$$L = \lambda_{2D} L_{2D} + \lambda_{3D} L_{3D}, \tag{4.12}$$

where λ_{2D} and λ_{3D} are the weights for 2D and 3D spatial losses, respectively.

4.2.3 Implementation

Figure 4.4 presents an overview of the network and loss function implementations and how back-propagation operates in the proposed RCNN. In the figure, we use two pairs of consecutive monocular and depth images for illustration. No ground truth poses are used for training. The transformation matrix $\hat{T}_{t->t+1}$ directly generated from the network and its inverse $\hat{T}^{-1}_{t->t+1}$ are used for loss calculation. Specifically, we use the monocular image I_t, depth information D_t at time t and the transformation matrix $\hat{T}_{t->t+1}$ to reconstruct the monocular image $\hat{I}_{t+1}$ at time $t{+}1$. Similarly, the reconstructed monocular image $\hat{I}_t$ at time t can be obtained by the monocular image I_{t+1}, depth information D_{t+1} at time $t+1$ and the inverse of the transformation matrix $\hat{T}^{-1}_{t->t+1}$. The 2D spatial loss L_{2D} can thus be calculated by Equation 4.8. We then use the depth information D_t at time t and the transformation matrix $\hat{T}_{t->t+1}$ to reconstruct the depth information $\hat{D}_{t+1}$ at time $t+1$. Similarly, the depth information $\hat{D}_t$ at time t can be obtained by

Algorithm 4.1: Implementations of the RCNN and loss functions.

```
Input  : Consecutive monocular images $\{I_1, I_2, ..., I_N\}$
         Associated depth images $\{D_1, D_2, ..., D_N\}$
Output: Trained RCNN
function prepare_Training_Data
    for $i$ in $(1 : N + 1)$ do
        if $i > (N_{seq} - 1)/2$ and $i < N - (N_{seq} - 1)/2$ then
            resize $I_i$ to $416 \times 128 \times 3$
            project Velodyne point cloud to depth image $D_i$
            resize $D_i$ to $416 \times 128 \times 1$
            stack $I_i$ and $D_i$ horizontally
            save camera intrinsics matrix file
        end
    end
    split data into two parts for training and testing
end
function build_Training_Graph
    prepare training data and camera intrinsics matrix path
    design data augmentation based on luminance $\gamma$, scale $s_x, s_y$ and rotation $r_d$
    design RCNN
    design total loss $L = \lambda_{2D} L_{2D} + \lambda_{3D} L_{3D}$
end
function Train
    load hyper parameters
    set $thres\_Epoch = 30$ based on experimental experience
    if epoch<thres_Epoch then
        feed training data into RCNN
        compute $L$
        adjust RCNN parameters
        if $step \% 500 = 0$ then
            collect summary
            save network
        end
    else
        break
    end
end
```

the depth information D_{t+1} at time $t + 1$ and the inverse of the transformation matrix $\hat{T}^{-1}_{t->t+1}$. The 3D spatial loss L_{3D} can thus be calculated by Equation 4.11. We then calculate the total loss L based on Equation 4.12. The total loss is then back-propagated through the network, adjusting its weights and making it closer to the truth in the next round. The orange arrows

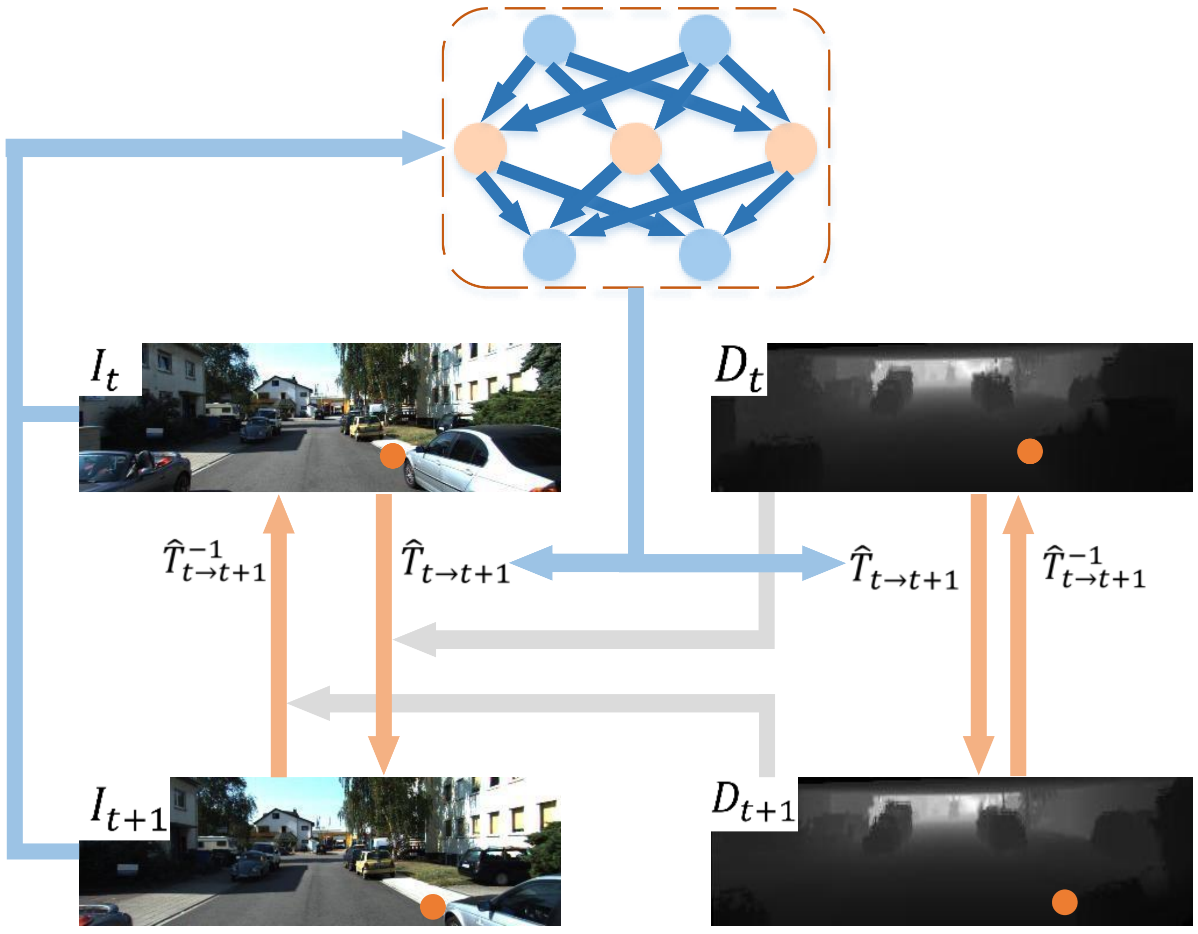


Figure 4.4: Training overview.

show how a pixel or a voxel can be projected to its neighbouring frame. Depth images are used for both 2D and 3D spatial loss calculation, thus the absolute scale can be recovered. Algorithm 4.1 presents a detailed implementation scheme.

4.3 Experiments

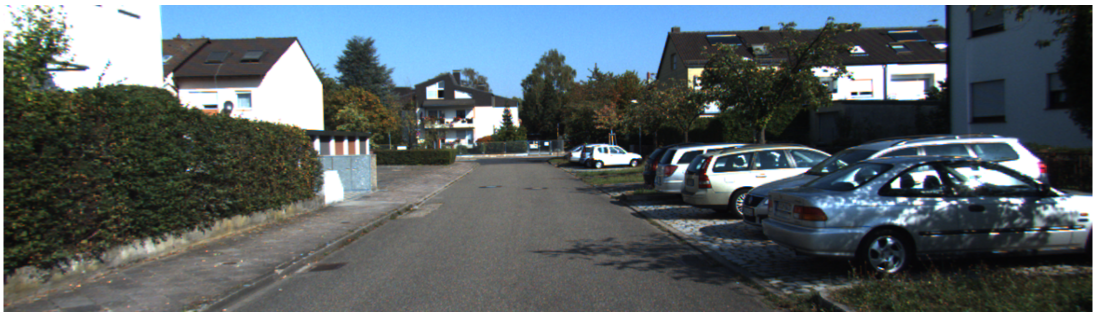
In this section, we first present the training details and then compare the performance of our VO system with other state-of-the-art algorithms in terms of both translation and rotation accuracy.

4.3.1 Training

We trained the proposed RCNN on a DELL workstation with an Intel Core i7-4790K @4.0GHz CPU and a Nvidia GeForce GTX Titan X 12GB Memory GPU. The model implementation environment is TensorFlow [169], which is an open source software library originating from

(a) Original monocular image without data augmentation.

(b) Luminance correction.


(c) Image rescale and cropping.

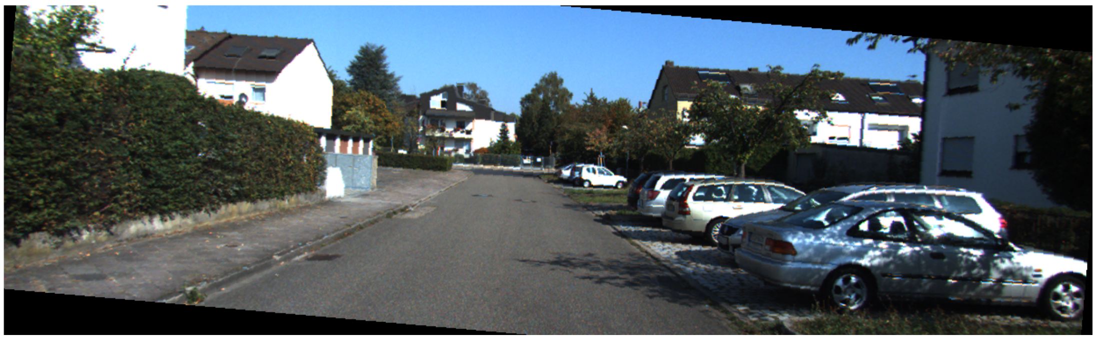

(d) Clockwise image rotation.

Figure 4.5: Data augmentation.

Google's Machine Intelligence research organization for numerical computation using data flow graphs. For fair comparison in Section 4.3.2, we adopted the same training dataset presented by

Zhou *et al.* [147] based on KITTI dataset only.

Before training, we resized the monocular images to 416×128 with 3 RGB channels and projected associated 3D point clouds to 2D single-channel depth images. Each point in a depth image represents the depth value of the corresponding point in the monocular image. Since KITTI data is relatively limited, online data augmentation technique is applied to enlarge the dataset and the results are shown in Figure 4.5. More specifically, the augmentation processing includes:

- Luminance: The input monocular images are randomly corrected by gamma $\gamma \in [0.7, 1.3]$.
- Scale: The input monocular and depth images are randomly scaled by scale factors $s_x \in [1, 1.2]$ and $s_y \in [1.0, 1.2]$ along X-axis and Y-axis, respectively. The images are then randomly cropped to 416×128.
- Rotation: The input monocular and depth images are randomly rotated by $r_d \in [-5, 5]$ degrees. Nearest-neighbour interpolation is used.

Note that the camera equipped on the KITTI car has a wider field-of-view than the LiDAR sensor, thus we only used the cropped region presented in [170] for loss calculation, as shown in Figure 4.6.

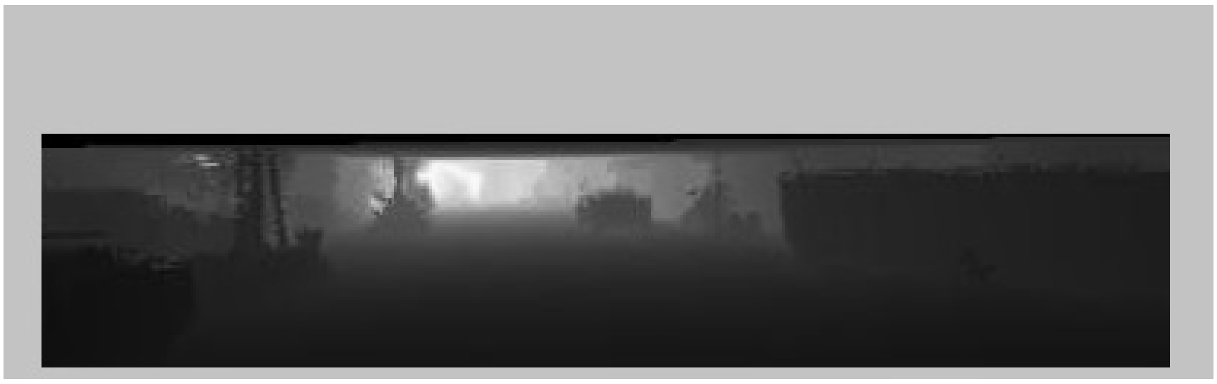


Figure 4.6: Region of interest for loss calculation. Ignored region is grayed out. X-axis: cropped from 15 to 401. Y-axis: cropped from 53 to 126.

We then fed pairs of monocular and depth images into the RCNN and trained the network from scratch. No ground truth poses were used during training. We employed the Adam optimization algorithm [171], which is an extension to Stochastic Gradient Descent (SGD) method and has recently been widely adopted in deep learning. Experiential parameters were used

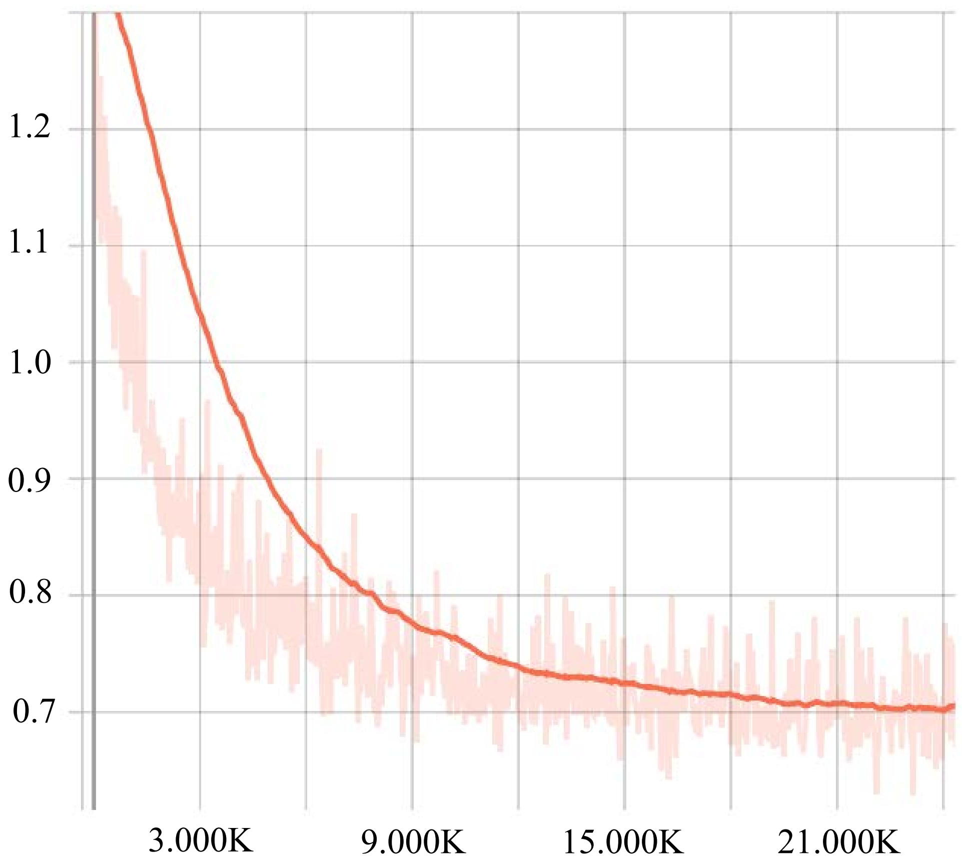
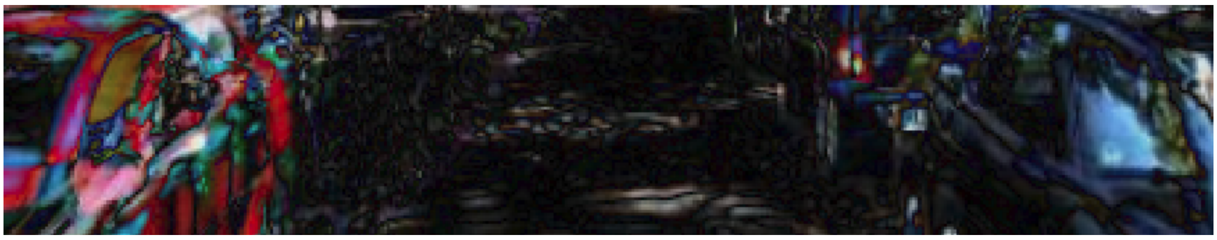
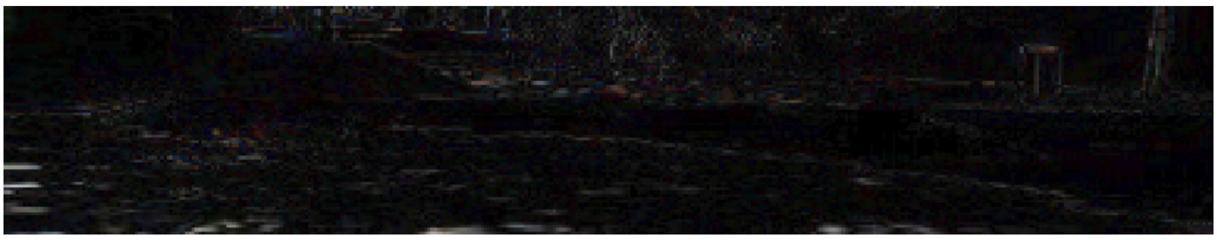


Figure 4.7: Training loss. X-axis: training steps. Y-axis: total loss.



(a) Disparity at the beginning of training.



(b) Disparity at the end of training.

Figure 4.8: Change of disparity during training.

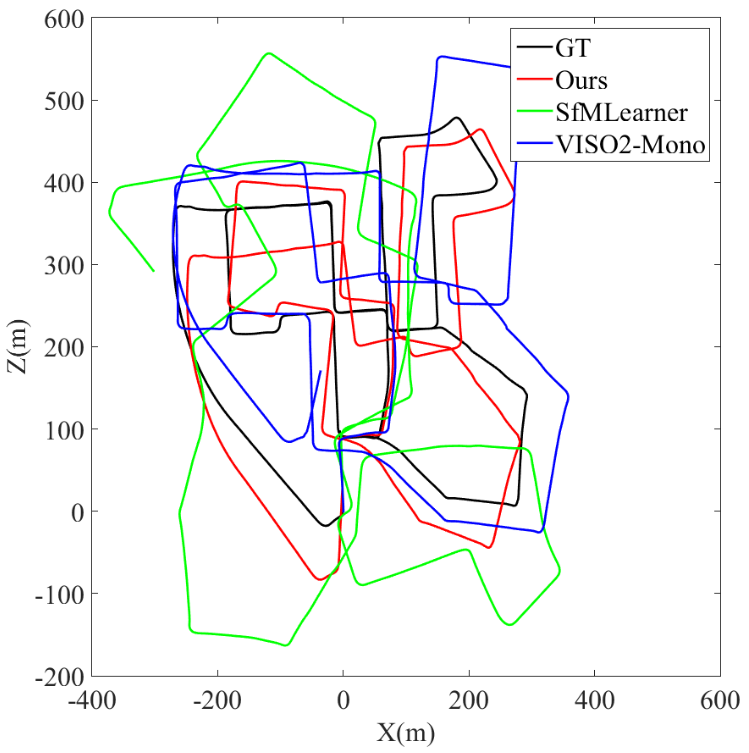
with the exponential decay rates for the first and second moment estimates being $\beta_1 = 0.9$ and $\beta_2 = 0.999$, respectively. We set $N_{seq} = 5$ and trained the RCNN for 40 epochs in total. The batch size is 32. Our learning rate schedule is step-decay based. The initial learning rate was set to 0.0002 and dropped to 0.0001 after 3/4 of the total training steps. No batch normalization was used since we found that it resulted in slow and unstable loss convergence in our experiments.

The total loss against training steps is shown in Figure 4.7. From the figure, we can see the total loss dropped rapidly before 9000 steps and then reduced slowly. Finally, it reached 0.7 at step 24000. At the same time, the disparity image between a monocular image and its projected image from its neighbouring frames and the pixel bar chart were used to visualize and monitor the training process. As can be seen from Figure 4.8, the images grew darker during training, i.e., the disparity narrowed.

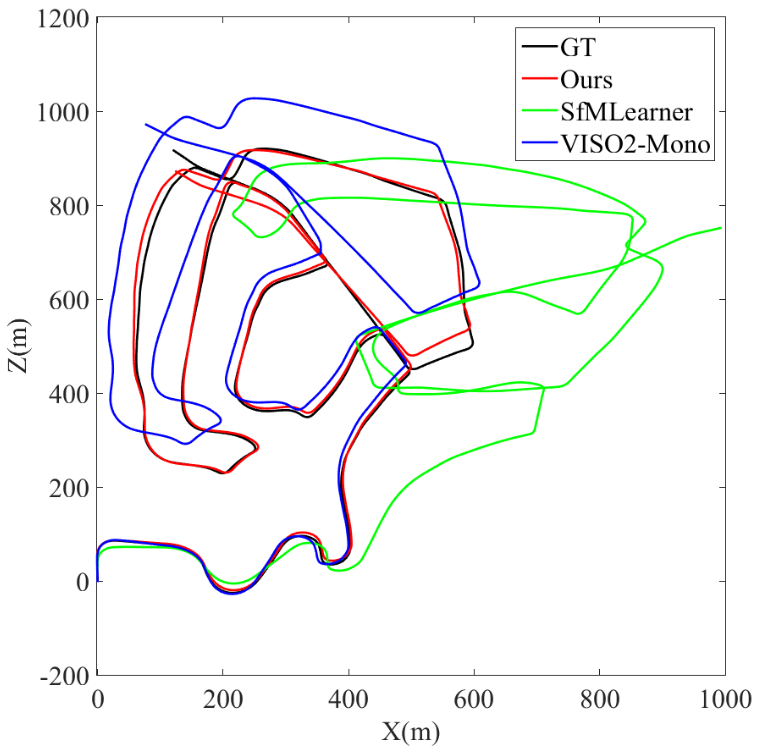
4.3.2 Performance Evaluation

Performance evaluation was carried out on a desktop computer with an Intel Core i7-3370 @3.4GHz CPU and a Nvidia GeForce GTX 980 4GB Memory GPU. Our proposed VO system was compared to other state-of-the-art VO systems based on KITTI Odometry dataset. The benchmark includes 22 stereo sequences and ground truth poses are provided for 00-10 sequences. The images were captured on a vehicle at 10 Hz which was moving in a city, rural areas and on highways at speed ranging from 0 km/h to 90 km/h. The scenes in the dataset are not static. Moving objects include cars and pedestrians. All these factors produce disturbance to VO systems and make the task more challenging.

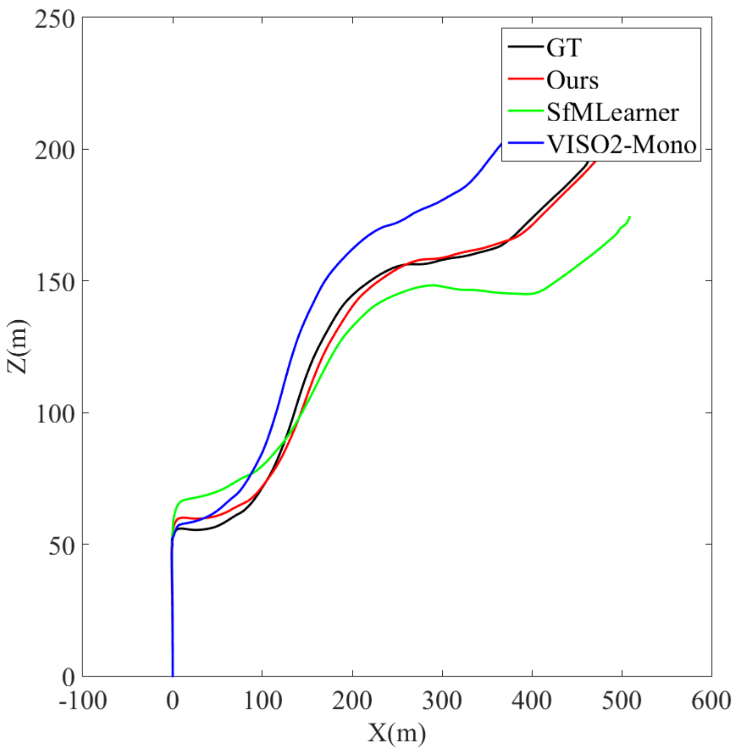
Our system can generate VO on an absolute scale without data post-processing. During testing, the network took only consecutive monocular images as input and directly generated poses. Thus our system is still a monocular VO system. We compare the proposed method to other state-of-the-art monocular VO systems, namely SfMLearner [147], VISO2-Mono. VISO2-Stereo [172], which is a stereo VO system is also used as a reference. No loop-closure detection (automatic or manual tagging) was applied and the same parameter set was used for all sequences. Our system and SfMLearning are unsupervised deep learning based, whereas VISO2-Mono and VISO2-Stereo are feature based. Since SfMLearning relies on ground truth poses for scale recovery, we post-processed the SfMLearning results for comparison. VISO2-Mono recovers absolute scale through a fixed camera height. VISO2-Stereo directly outputs poses on an absolute scale since it employs stereo sequences for testing. The input image resolution of our system and SfMLearning is 416×128, whereas VISO2-Mono and VISO2-Stereo



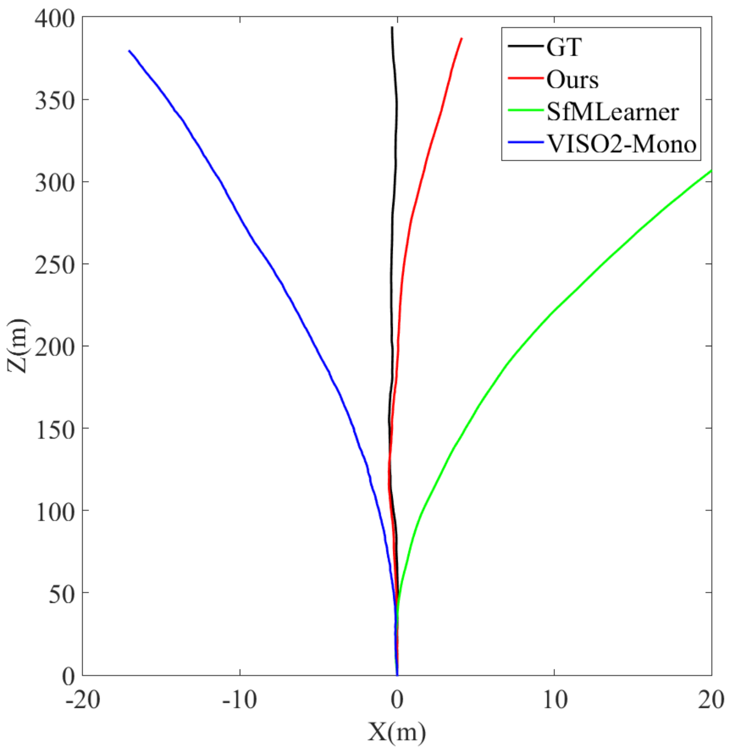
(a) Sequence 00.



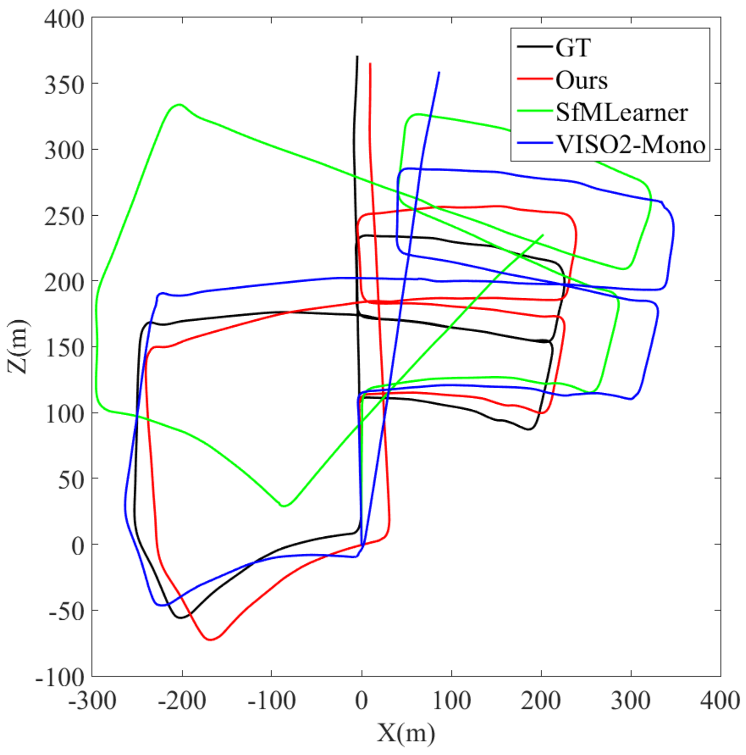
(b) Sequence 02.



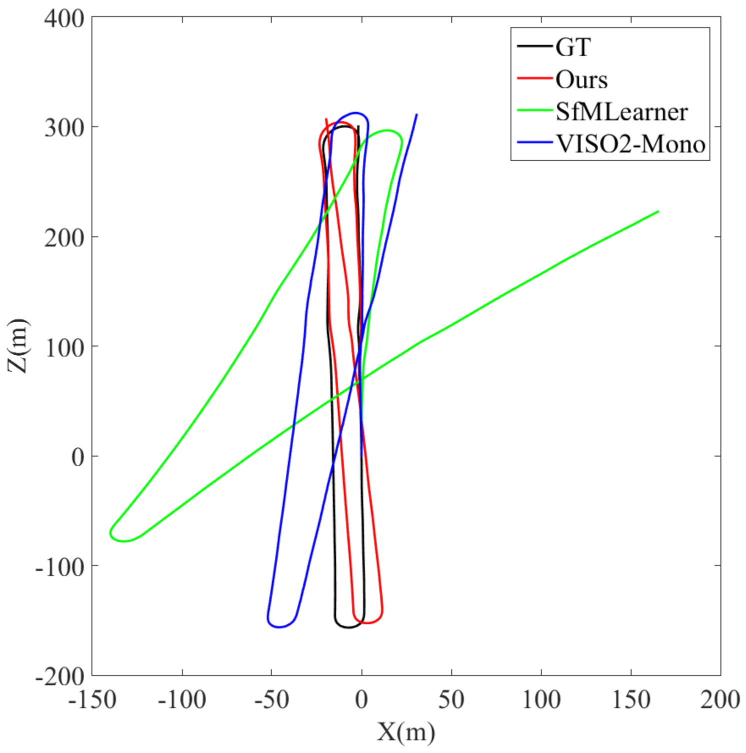
(c) Sequence 03.



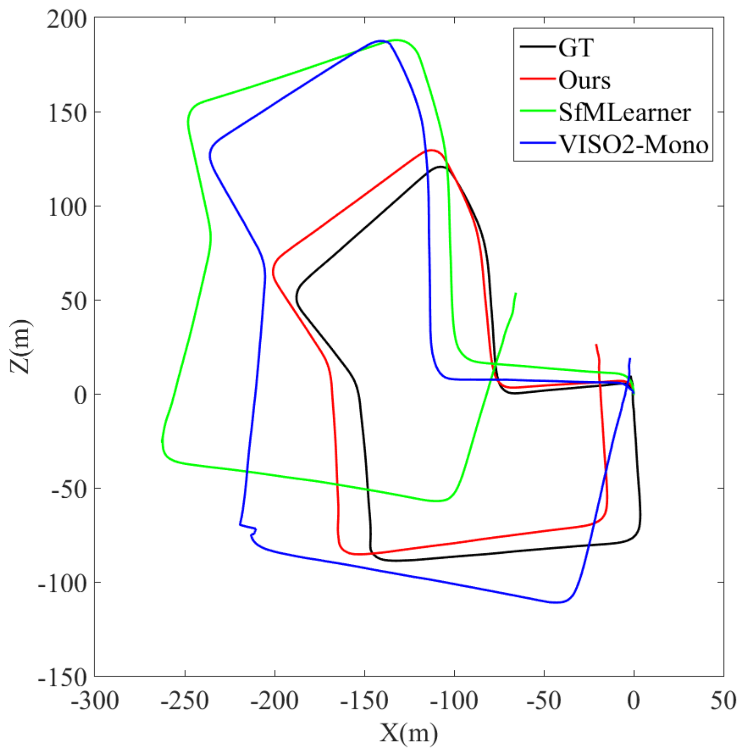
(d) Sequence 04.



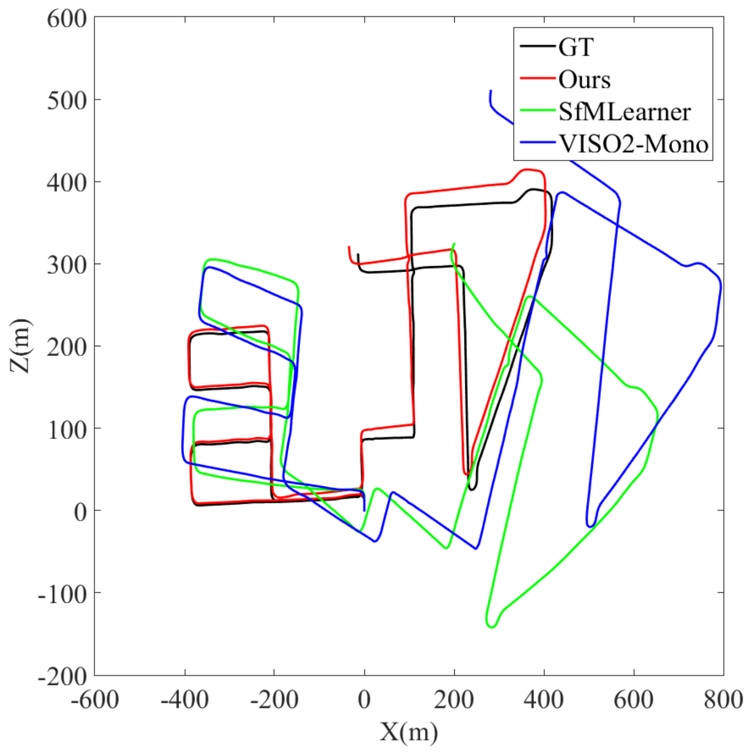
(e) Sequence 05.



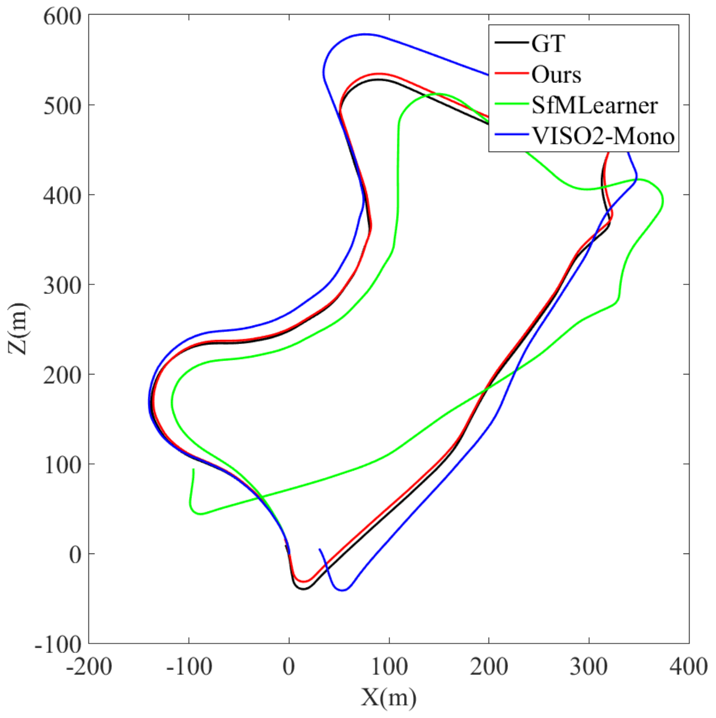
(f) Sequence 06.



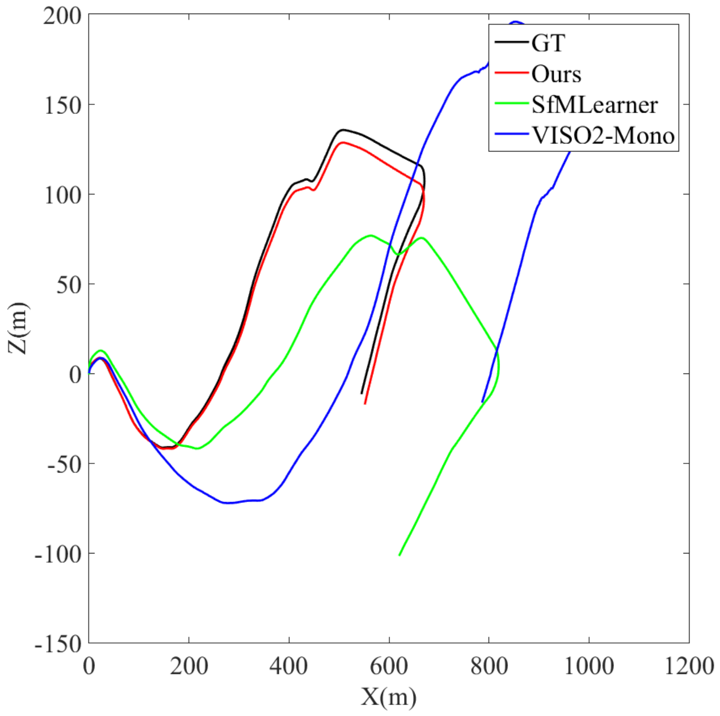
(g) Sequence 07.



(h) Sequence 08.



(i) Sequence 09.



(j) Sequence 10.

Figure 4.9: Trajectories of KITTI Odometry Sequence 00 and Sequence 02-10.

adopt a 1242×376 setting.

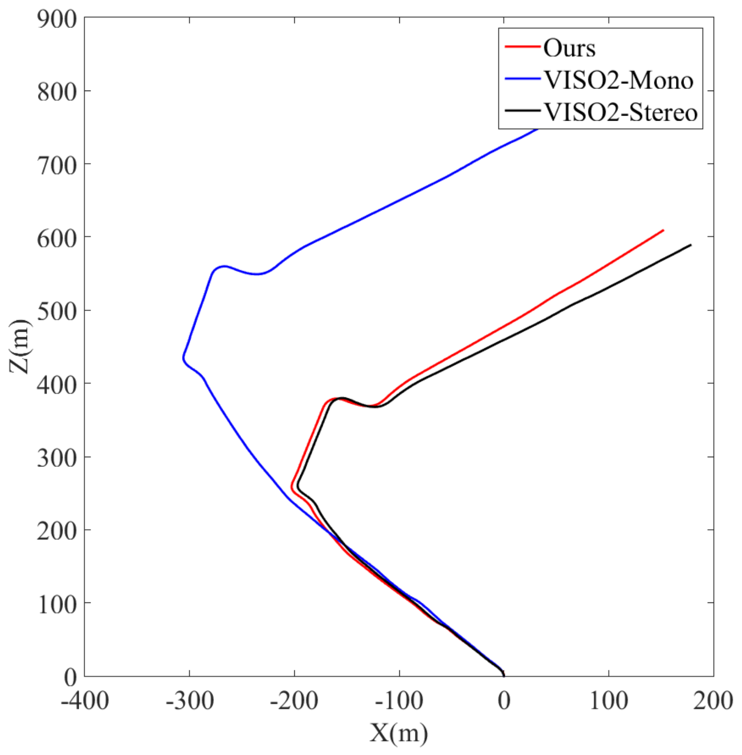
Figure 4.9 shows the trajectories of KITTI Odometry from Sequence 00 to Sequence 10. The results were generated by our system, SfMLearner and VISO2-Mono. Ground truth trajectories are provided as a reference. Sequence 01 is omitted since the sequence was captured on a highway with rare features. Thus, all methods including VISO2-Stereo failed to recover the absolute scale. SfMLearner results are post-processed with ground truth poses for scale recovery. Only 2D trajectories (X-axis and Z-axis) are provided for clearer presentation. The vertical Y-axis is omitted. As can be seen from Figure 4.9, the proposed method outperforms other monocular VO systems. The generated trajectories are the closest to the ground truth ones. VISO2-Mono generally performs better than SfMLearner because it recovers absolute scale through a fixed value. SfMLearner has to post-process each transformation matrix with ground truth to obtain absolute scale. Thus, using a fixed value reduces its performance. All algorithms perform well on Sequence 03, 09 and 10, as shown in Figure 4.9c, 4.9i and 4.9j. This is because the car speed changes are small. Thus the trajectories can be well generated even a constant value is used to recover absolute scale. From Figure 4.9a, 4.9b and 4.9f, we can tell SfMLearner starts to perform much worse than the other two algorithms because the car speed changes become bigger. VISO2-Mono performs bad on Sequence 04 and 08, as shown in Figure 4.9d and 4.9h, because the deviations of the estimated rotation matrices are too large, whereas Figure 4.9g shows there are significant deviations of the estimated translation matrices. All methods perform bad on Sequence 05 due to shape turns and hard brakes. It should be noticed that Sequence 09 and 10 are not used for training. However, the results of these two sequences show the pre-trained model can be generalized and applied to other similar scenes.

The detailed translational and rotational errors are listed in Table 4.2. We adopt Root Mean Square Error (RMSE) recommended by KITTI for evaluation. The translational errors are measured in percent $(\%)$, whereas the rotational errors are measured in degrees per meter $(^\circ/m)$. Each value in the table was obtained by averaging errors of all possible subsequences of length 100, 200,..., 800 meters. From the table we can see our method generated lower errors than other monocular systems in terms of both translation and rotation and can be compared to a stereo VO system. We can further reduce the rotational errors by manually increasing the ratio of the training images captured when the vehicle is turning. Since KITTI dataset is relatively small, the overall performance of our network can also be improved by employing larger dataset

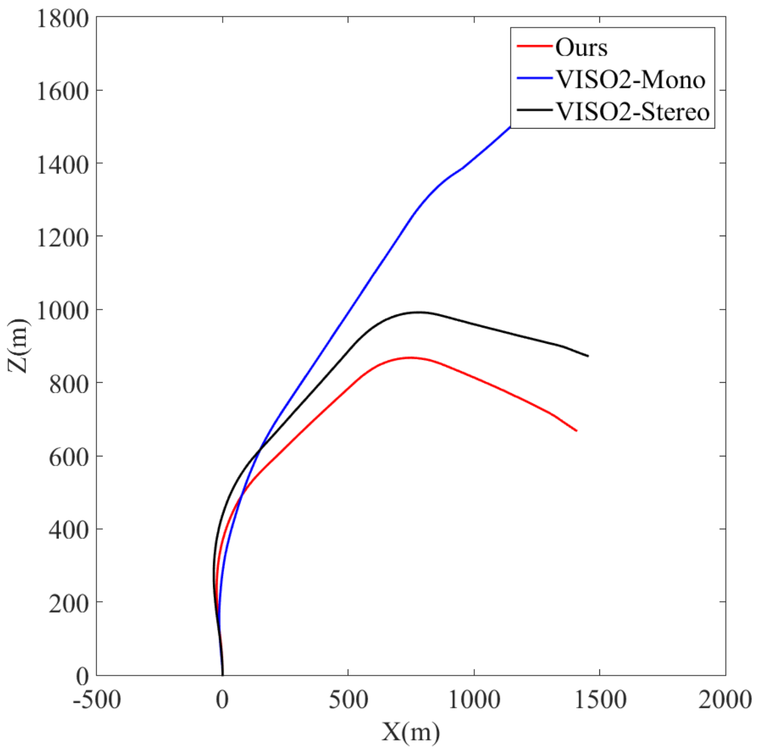
Table 4.2: Translational and rotational errors. VISO2-Stereo results are provided as a reference. SfMLearner poses are post-processed with ground truth poses for scale recovery. Sequence 09 and 10 are not used for training.

	Ours		**SfMLearner**		**VISO2-Mono**		**VISO2-Stereo**	
	Monocular		Monocular		Monocular		Stereo	
	Absolute Scale		Scale Post-processing		Absolute Scale		Absolute Scale	
	Unsupervised		Unsupervised		Feature based		Feature based	
Seq.	t_{rmse}	$r_{rmse} \times 100$	t_{rmse}	$r_{rmse} \times 100$	t_{rmse}	$r_{rmse} \times 100$	t_{rmse}	$r_{rmse} \times 100$
00	5.14	2.13	45.89	6.23	18.24	2.69	1.86	0.58
02	4.88	2.26	57.59	4.09	4.34	1.18	2.01	0.40
03	6.03	1.83	13.08	3.79	8.47	1.96	3.21	0.73
04	2.15	0.89	10.86	5.13	4.69	1.80	2.12	0.24
05	3.84	1.29	16.76	4.06	19.22	3.54	1.53	0.53
06	4.64	1.21	23.53	4.80	7.30	1.78	1.48	0.30
07	3.80	1.71	17.52	5.38	23.61	4.11	1.85	0.78
08	2.95	1.58	24.02	3.06	24.18	2.47	1.92	0.55
09	5.59	2.57	22.27	3.62	7.08	1.15	1.99	0.53
10	4.76	2.95	14.36	3.98	41.60	3.03	1.17	0.43
mean	4.38	1.84	24.59	4.41	15.83	2.37	1.91	0.51

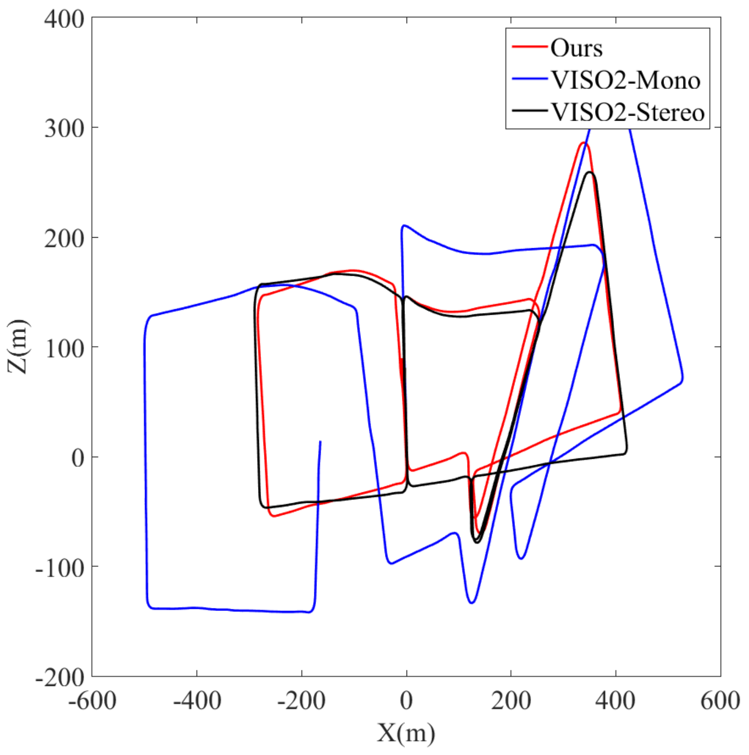
- $t_{rmse}(\%)$: average translational RMSE drift (%) on length of 100m-800m.
- $r_{rmse}(°/m)$: average rotational RMSE drift ($°/m$) on length of 100m-800m.



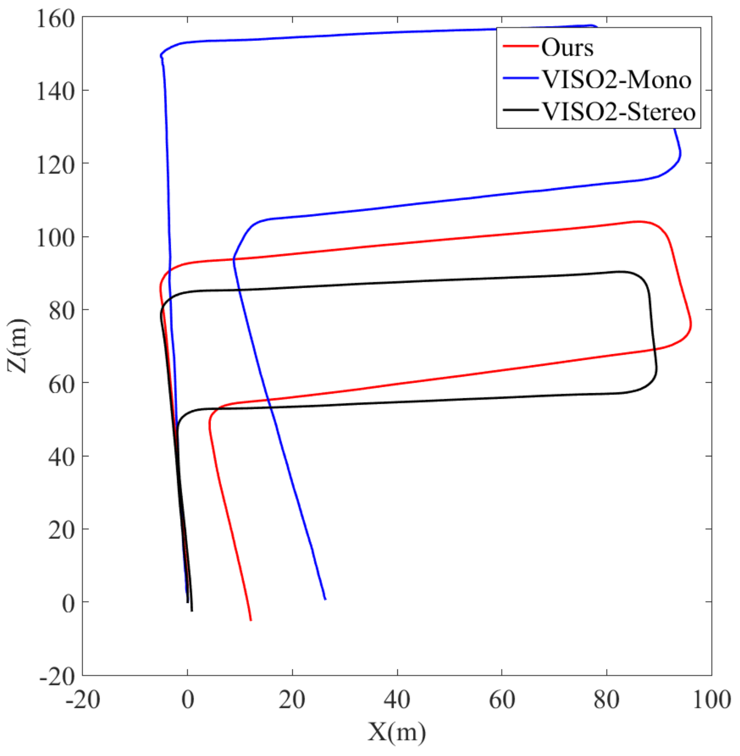
(a) Sequence 11.



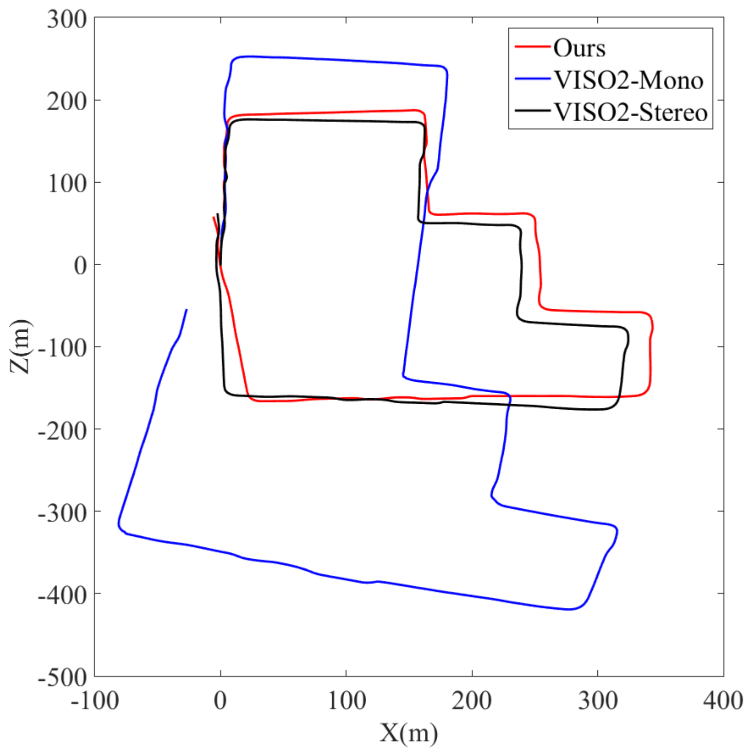
(b) Sequence 12.



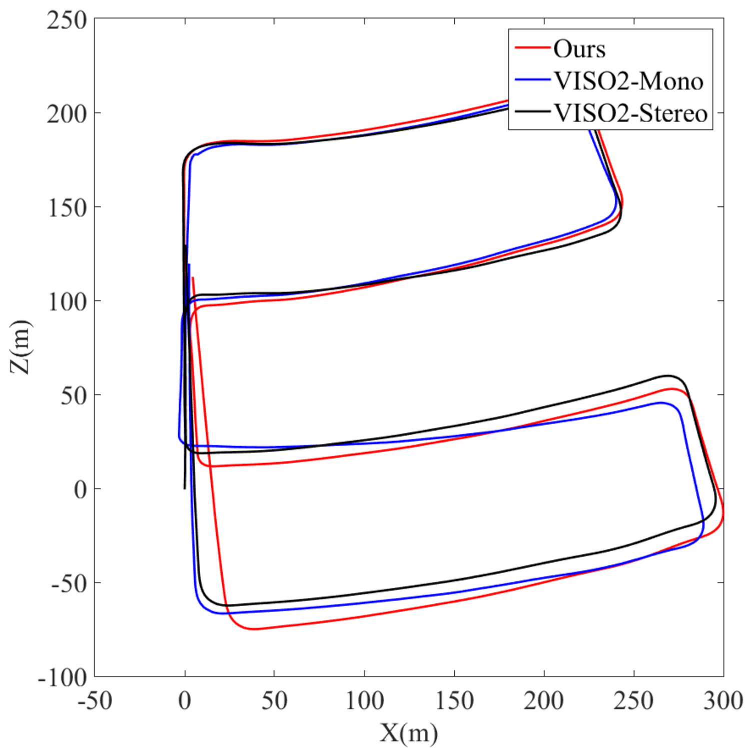
(c) Sequence 13.



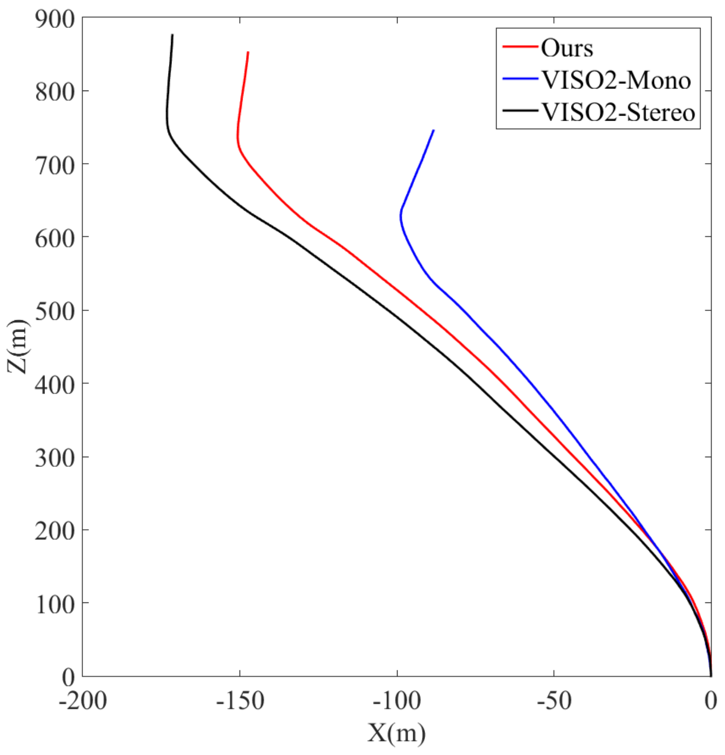
(d) Sequence 14.



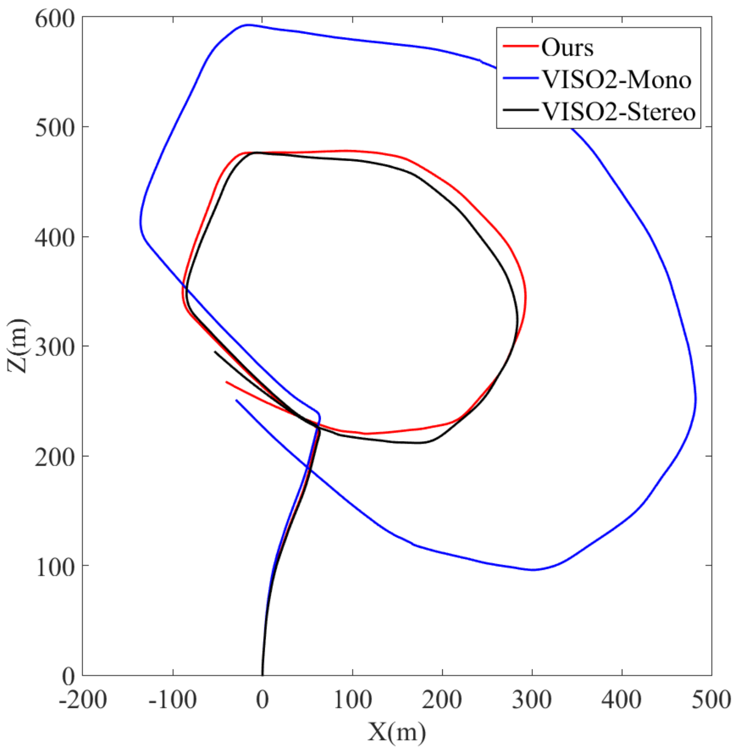
(e) Sequence 15.



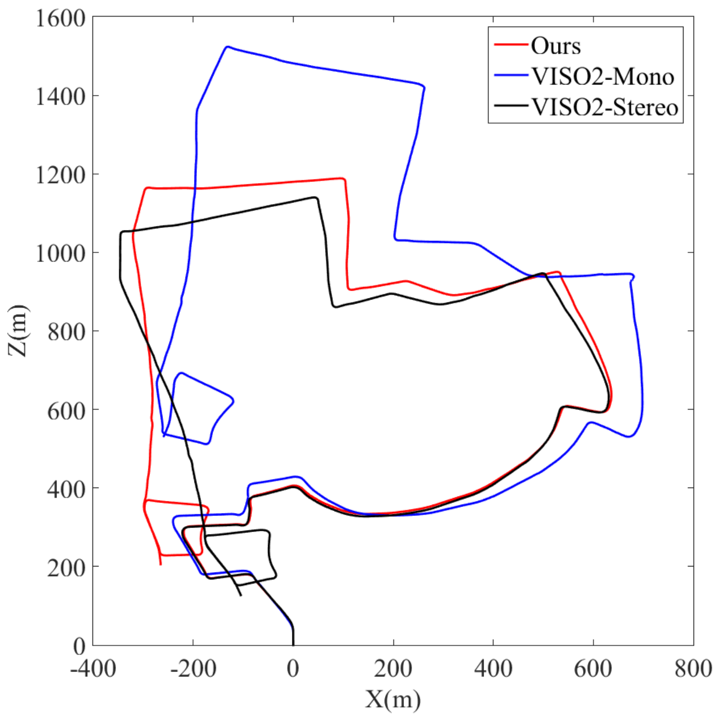
(f) Sequence 16.



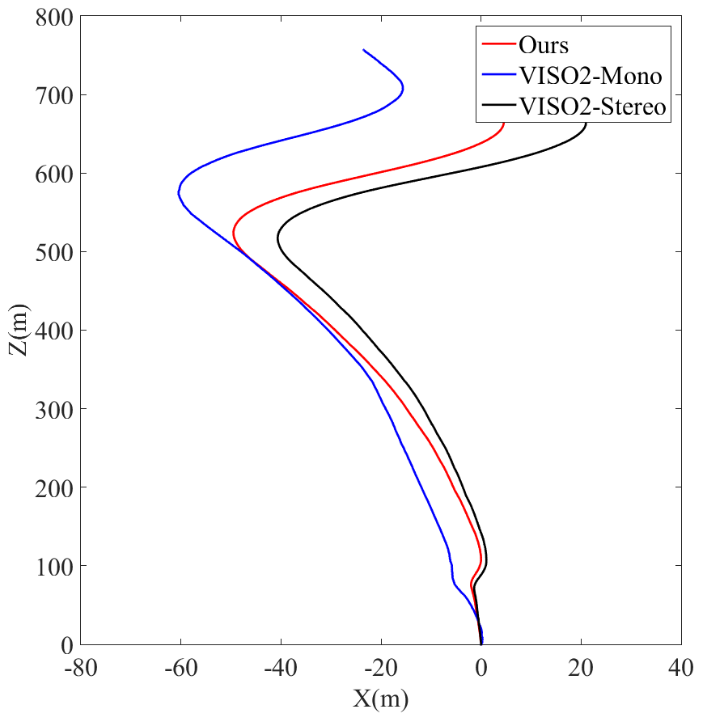
(g) Sequence 17.



(h) Sequence 18.



(i) Sequence 19.



(j) Sequence 20.

Figure 4.10: Trajectories of KITTI Odometry Sequence 11-20.

for training.

The trajectories of KITTI Odometry from Sequence 11 to 20 are presented in Figure 4.10. Only 2D trajectories (X-axis and Z-axis) are provided for clearer presentation. The vertical Y-axis is omitted. VISO2-Stereo trajectories are also provided as a reference. No ground truth poses are provided for these sequences. Thus, quantitative evaluation can not be carried out. We can see the performance of our method is close to VISO2-Stereo (our trajectories are closer to VISO2-Stereo than VISO2-Mono). Figure 4.10f and 4.10j show that both our method and VISO2-Mono perform well on Sequence 16 and 20. Similar to the previous discussion, the deviations of the estimated rotation and translation matrices inevitably exist due to significant speed changes and sharp turns. In addition, in the case when the car bumps, the distance from the camera to the ground changes. Therefore, using a fixed value to recover absolute scale is not reliable. From the figures we can tell our method generally outperforms VISO2-Mono in terms of translation estimation (Figure 4.10a, 4.10d, 4.10e, 4.10g, 4.10h) and rotation estimation (Figure 4.10b, 4.10c, 4.10i).

Although the proposed method outperforms other monocular VO systems in terms of translation and rotation accuracy, the processing time is longer. We set batch size to 1 for pose generation. The processing time is 0.09 second per pose based on a Nvidia GeForce GTX 980 GPU and the input image size being $416 \times 128 \times 3$, whereas VISO2-Mono and VISO2-Stereo systems require only a CPU to achieve a similar speed. Compared to other deep learning based methods, we require no ground truth poses for training or scale post-processing, but still need depth information for injecting the scale.

4.4 Summary

This chapter proposed a monocular visual odometry system based on deep learning technique. The system operates in an unsupervised end-to-end training manner. Consecutive monocular images and depth information are used for training. As no ground truth pose labeling is needed, the proposed system requires less human effort and is cheap to run. For testing, the proposed system takes only monocular images as input and directly generates poses on an absolute scale. Experiments were carried out on KITTI dataset. Results have shown that our system outperforms other monocular VO systems in terms of translation and rotation accuracy and can be compared to stereo VO systems. The pre-trained model can also be generalized to other scenes.

The performance of the system can be improved by further training.

The proposed method requires high computing power and is difficult to achieve real-time performance. Computational efficiency based on such unsupervised training manner can be further improved. Depth information can also be incorporated during testing in order to boost system performance for real-time navigation of autonomous robots and the visual guidance of blind people.

Chapter 5

Indoor Topological Localization Based on Deep Learning Technique

This chapter presents a novel localization approach to support the indoor localization of people with vision impairment or robots, which is based on Deep Convolutional Neural Networks. More specifically, a 3D indoor semantic map is firstly constructed using an RGB-D sensor, and the constructed map is then deployed to help users conducting indoor topological localization. The semantic information extracted from the mapping process can be used to diagnose and eliminate errors and boost topological localization performance despite appearance changes within the environment. Experiments are conducted to verify that the proposed methods can increase both precision and recall rates.

5.1 Introduction

Traditional maps built by robotic systems are either geometric or topological, which are navigation oriented and serve obstacle avoidance and path planning well [28]. However, these maps are passive and cannot provide useful semantic information to visually impaired people or service robots for assistance. Semantic information interpreted from scenes should also be included into the map to form a semantic map and provide a friendly human-machine interface [21]. In other words, a semantic map contains linguistic words representing places, landmarks and daily objects, which are very useful to assist human users.

This chapter first builds a 3D indoor geometric map using an RGB-D sensor and off-the-shelf algorithms. To incorporate semantic information, we adopt deep Convolutional Neural Networks (CNNs) for daily object detection, rather than the bag-of-visual-words model (BoW) which is commonly deployed by the SLAM community in recent years. In the case of CNNs, models trained for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) can classify the entire images into as many as 1,000 classes with acceptable error rate [173]. This enables the objects to be detected in a much broader scope. Moreover, objects in any shape, form or colour can be classified into one category as long as they have the same name. Thus, only object names are stored in our database rather than images, which results in a relatively small database. Once objects are detected, the subordination of the objects to rooms is then represented by the anchoring method. We then use the acquired semantic map for indoor topological localization of visually impaired people, i.e., answering their questions such as "Where am I?" or "Which room am I in?".

Generally speaking, constant changing appearance can be a significant factor in visual localization failure [108]. For instance, the lighting conditions vary between day and night, and objects (chair, laptop, curtain or even human) may be in different positions randomly. To address these problems, this chapter adds additional semantic data into the maps to improve localization performance. Since object recognition can aid place recognition [24, 25, 26], we use distinctive objects detected and labeled from the mapping process, as well as the undetected objects that might exist and can infer the function of a room such as "bedroom" or "lab", to further verify the locations. Experiments with long-term operation have also been carried out and shown how semantic information accounts for appearance changes within the environment. Compared to other state-of-the-art algorithms, our method generates higher precision and recall rates (recall is an essential factor for other tasks e.g. online map updating).

In the case of environment construction, most methods are feature based [5, 136]. Newcombe *et al.* [134] presented a system for real-time indoor mapping with a low-cost, lightweight Kinect camera. Individual observations from the depth stream were fused into a single global surface model. Mur-Artal *et al.* [155] employed a monocular camera for lifelong mapping. The construction operation recovers if the scene content changes. Thanks to the binary ORB feature, this system is highly computationally efficient even without GPUs. Other researchers deployed both feature points and plane patches for robust pose estimation [174]. Since SLAM is not our focus in this chapter, we adopt an off-the-shelf feature based method for environment construction.

Semantic information can be extracted from both range and visual sensors. In indoor environments, architectural components and room functions were inferred from laser data by Nüchter [30] and Mozos [129] *et al.* respectively. Attentions were then drawn to visual sensors due to the availability of richer features. Grimmett *et al.* [49] presented a framework which can extract driving lanes and parking spaces for vision-only automated parking. Random Forests (RFs) were applied [136] to construct a consistent 3D indoor model from 2D semantic segmentations. BoW approach was also used for object detection [175] by building SURF keypoint and local colour histogram dictionaries. In this chapter, room functions and objects are considered as semantic information.

Topological localization tackles the problem of recognizing the place when we revisit a scene. Ulrich and Nourbakhsh [47] built colour histograms of the acquired images, each of which voted for the most likely location in the database. Wang *et al.* [176] used an additional SIFT descriptor to index the images in the database for image retrieval. BoW model have been widely used for loop closure detection (detecting already-mapped scenes) during SLAM [177, 178, 143, 179]. Inspired by document classification, BoW represents images by the occurrence frequency of individual hand engineered features in each dictionary. The performance is highly dependent on the features selected.

Convolutional Neural Networks (CNNs) have recently been widely used as robust visual feature extractors in the computer vision and machine learning domains and have shown better performance in terms of changing environments, viewpoints, lighting conditions, objects, etc. [103, 104]. Sharif Razavian *et al.* [104] have shown that CNNs outperform BoW in most recognition tasks in terms of large datasets.

Although most CNNs are trained for object recognition, some researchers have managed to modify these models for other related but different tasks such as place recognition and object detection [104, 105, 106] since the generic features learned by different models from holistic images in different datasets are versatile and transferable [107, 108]. In this chapter, we adopt the Inception-v3 model [180] for place recognition and object detection.

Researchers have already shown that place recognition can benefit from object recognition [24, 25, 26], especially in indoor environments where a location can be revealed by the detected objects. However, if a recognition method only relies on objects, it fails in the case that no distinctive objects are spotted within the camera's field of view. Moreover, some objects (curtain, sink, computer, etc.) are not sufficient on their own to infer precise locations. Thus, we combine object detection with a holistic approach for better localization performance.

The rest of the chapter is organized as follows. Our semantic information extraction and topological localization method are detailed in Section 5.2. Training and experiment results are subsequently presented and discussed in Section 5.3. Finally, a brief conclusion is given in the last section.

5.2 Preliminaries

This section explains our semantic mapping and topological localization methods. We use the same environment construction methods presented in Chapter 3. The traditional geometrical mapping method is used, which includes feature extraction and representation, feature matching, transformation estimation, loop closure detection and global pose graph optimization. In the case of semantic information extraction, we use a deep learning based method rather than the feature matching based method presented in Chapter 3. The semantic mapping approach used in this chapter is shown in Figure 5.1 with orange boxes. Figure 5.3 shows a constructed 3D environment map for an office floor at Essex University and Figure 5.2 shows a constructed environment map for a flat where students live.

5.2.1 Semantic Information Extraction and Representation

The semantic information used in this section consists of indoor places and the daily objects within them. The place names are hand-coded into the database, whereas object names are extracted from the aforementioned keyframes. An indoor place constructed by hundreds of keyframes normally contains various objects. If all objects in each keyframe are identified and labeled, our database would be intractable.

In fact, we are more interested in distinctive objects which can infer the function of a place. Moreover, detecting object in a single image inevitably generates errors. Thus, the following rules are used for object detection.

- Only one object can be identified from each keyframe.
- The output score of the detected object needs to exceed over a threshold.

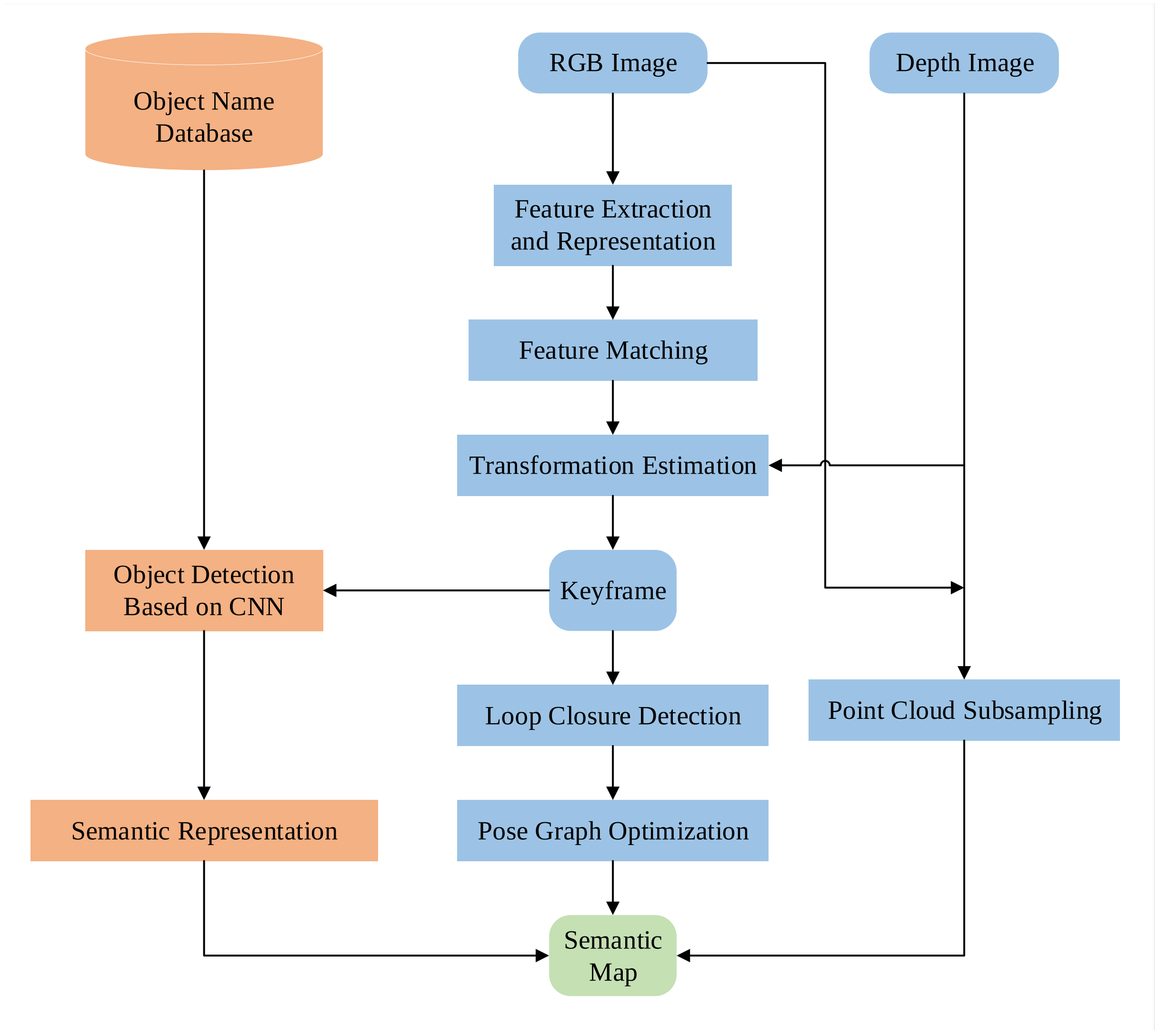


Figure 5.1: Semantic mapping process. Blue boxes: 3D scene construction process. Orange boxes: semantic information extraction process.

- An object can be labeled only if it has been detected in 15 continuous keyframes.

Therefore, semantic information extraction becomes an object recognition problem since it only involves stating whether an image contains a specific object, not the position of the object inside the image. Thus, a pre-trained Inception-v3 model [180] is deployed. The model trained for ImageNet competition can classify objects into 1,000 categories, which is powerful enough for our system.

Finally, the conceptual knowledge is represented with "has-a" relations [75], as shown in Figure 5.4. The straight lines from left to right indicate this relationship, and conversely, the objects on the right can reveal the associated locations on the left. Note that some objects in our

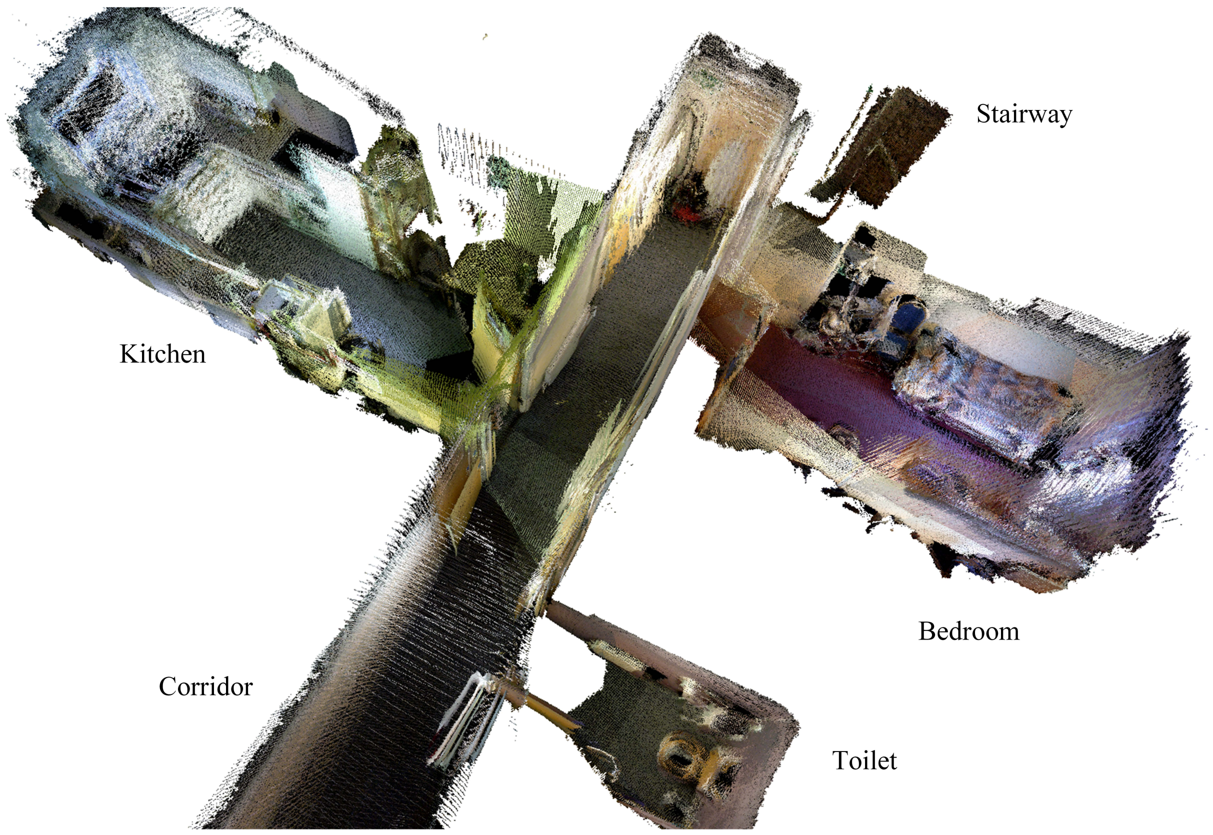


Figure 5.2: An environment map for a flat

database can infer multiple locations.

5.2.2 Topological Localization

In this section, we explain the proposed topological localization methods in detail. The block diagram of this method is shown in Figure 5.5. Although in some cases, distinctive objects in an indoor environment can directly tell us locations, it is still necessary to deploy a holistic approach since not all observations contain these objects. In addition, some objects can be discovered in multiple places. Therefore, relying entirely on object detection for localization is impractical. Thus two Convolutional Neural Networks are trained beforehand and used in this chapter. Algorithm 5.1 presents a detailed topological localization scheme.

Once a query image is received, we directly feed the holistic image into two CNNs for place recognition and object detection respectively. A place recognition result consisting of the predicted score for each location can be firstly obtained. If the object detection score is over a threshold and a distinctive object in the database can be found, we use this additional semantic information to rectify the place recognition predicted scores, which can be viewed as a post-

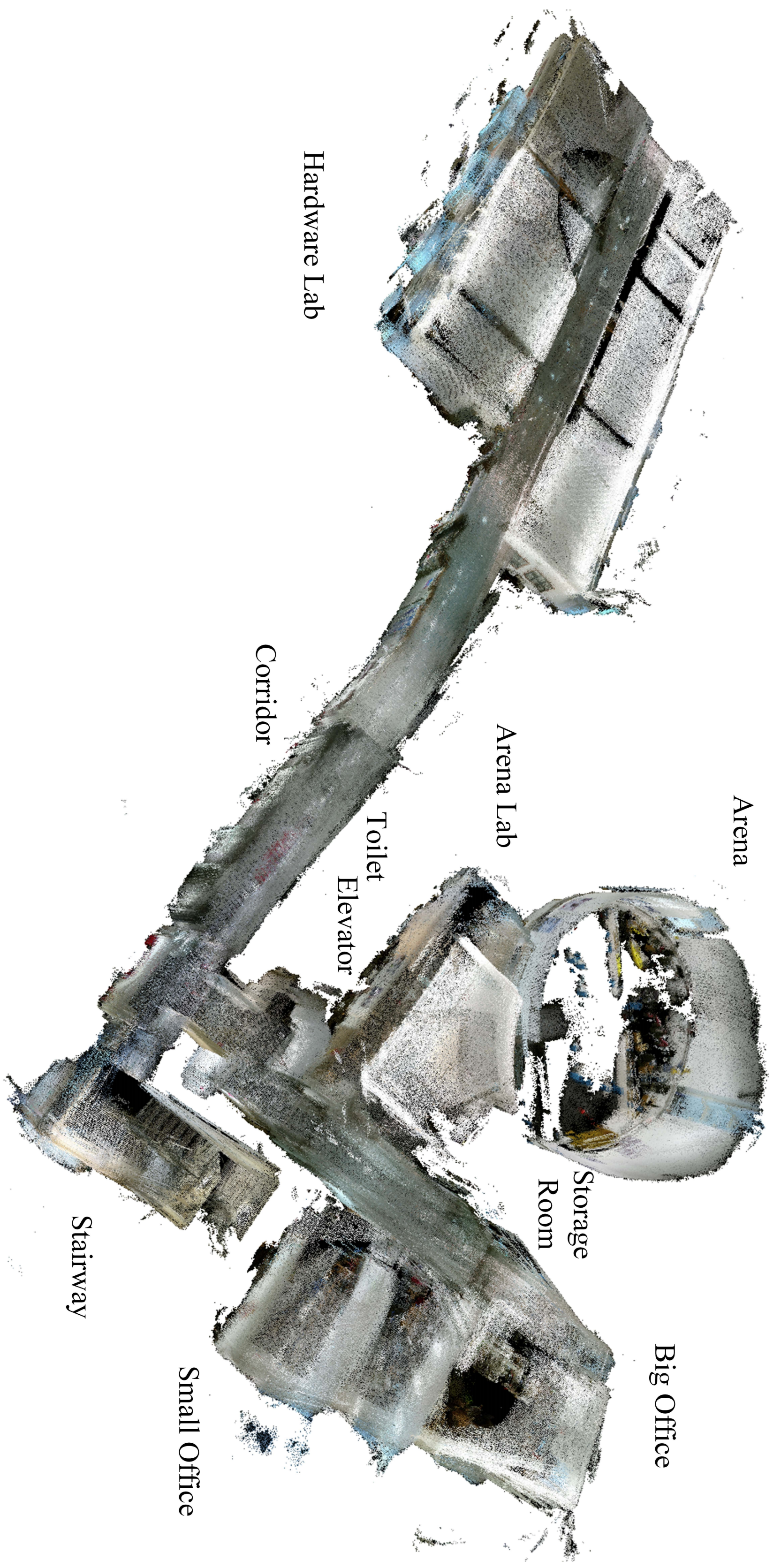


Figure 5.3: An environment map for an office floor

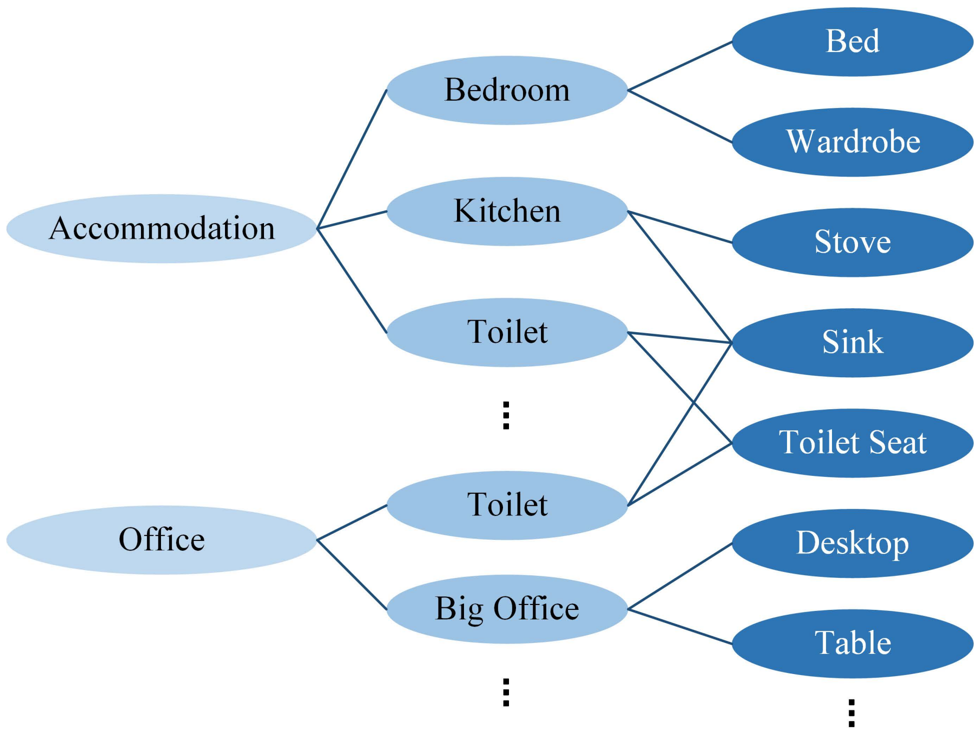


Figure 5.4: Semantic information representation.

processing step, otherwise the place recognition result is taken as the localization final result. The threshold is discussed in the experiment section.

Researches have shown that the generic features learned from different CNNs are transferable. One CNN can be retrained and used for different recognition tasks. Therefore, we adopt the Inception-v3 model [180] for both place recognition and object detection. The training methods are detailed in the next section.

If a distinctive object in the database is detected, we then simply use a Bayesian approach to rectify the place recognition predicted scores. Let $\boldsymbol{L}$ be the location vector

$$\boldsymbol{L} = \{l_1, l_2, \ldots, l_n\}, \tag{5.1}$$

where n is the total number of locations in the database, l_i is the category index of location i. Given a query image x with object detected within it, the basic Bayesian inference is applied to estimate the rectified score $P(l_i|x)$

$$P(l_i|x) = \frac{P(l_i)P(x|l_i)}{P(x)}, \tag{5.2}$$

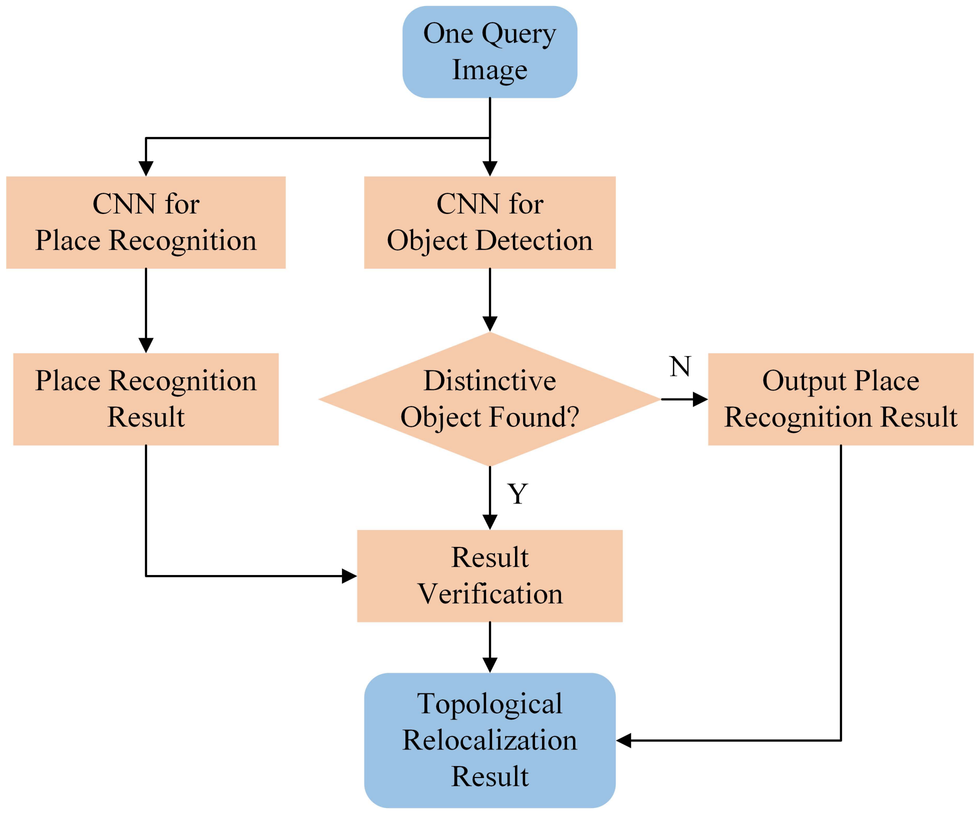

One Query Image
CNN for Place Recognition
CNN for Object Detection
Place Recognition Result
Distinctive Object Found?
N
Output Place Recognition Result
Y
Result Verification
Topological Relocalization Result

Figure 5.5: Topological localization method with one query image.

Algorithm 5.1: Topological localization scheme.

```
Input  : Consecutive monocular images $\{I_1, I_2, ..., I_N\}$
Output: Location
function generate_Location
    load hyper parameters and network parameters for both CNNs
    for $j$ in $(1 : N + 1)$ do
        feed $I_j$ into the first CNN to compute place recognition score $P(l_i)$
        feed $I_j$ into the second CNN to compute object detection score $P_o$
        set $thres\_Object\_Detection = 0.7$ based on experimental experience
        if $P_o$>$thres\_Object\_Detection$ then
            compute $P(l_i|x)$
            return inferred location based on $P(l_i|x)$
        else
            return inferred location based on $P(l_i)$
        end
    end
end
```

where $P(l_i)$ is the place recognition predicted score, $P(x)$ is the probability of the object existing in the image, $P(x|l_i)$ is the empirical knowledge. Since the denominator $P(x)$ is identical to all locations, we have

$$P(\boldsymbol{L}|x) \propto P(\boldsymbol{L})P(x|\boldsymbol{L}), \tag{5.3}$$

in which $P(x|\boldsymbol{L})$ is the empirical probability distribution. Finally, we output the normalized distribution $P(\boldsymbol{L}|x)$ as the final topological localization result.

Each object in the database has its own empirical probability distribution in terms of all locations. Based on the semantic representation created in the mapping process, assume $\boldsymbol{L}_w = \{l_1, l_2, \dots, l_p\}$ is a set of locations with a specific labeled object y in them, whereas $\boldsymbol{L}_{wo} = \{l_1, l_2, \dots, l_q\}$ is a set of locations without this object in them. A ratio is used to obtain the distribution by

$$\xi = \frac{P(y|l_r)}{P(y|l_s)}, \tag{5.4}$$

in which ξ is a given factor, $l_r \in \boldsymbol{L}_w$, $l_s \in \boldsymbol{L}_{wo}$.

The factor ξ plays an important role in our system. It controls the weights of the two CNN stream outputs. On one hand, we want a large factor so that the system still performs well even though a location suffer from changing appearance or human intervention (Figure 5.6a and Figure 5.6b, sliding door detected). However, the precision drops if the factor is too large since object detection errors inevitably exist. Furthermore, some objects randomly appeared in other locations where they should not belong to would also lead to localization errors. For example, although a vacuum is found in Figure 5.6d, the location should still be labeled as accommodation corridor rather than storage room. The factor ξ is further discussed in Section 5.3

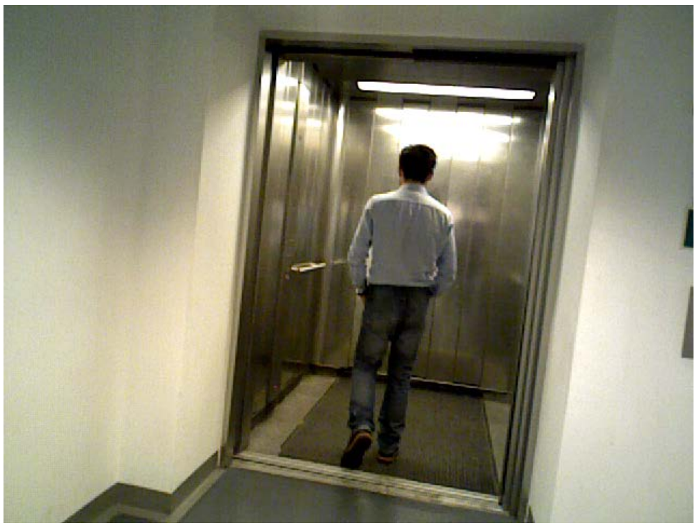
5.3 Experiments

5.3.1 Training

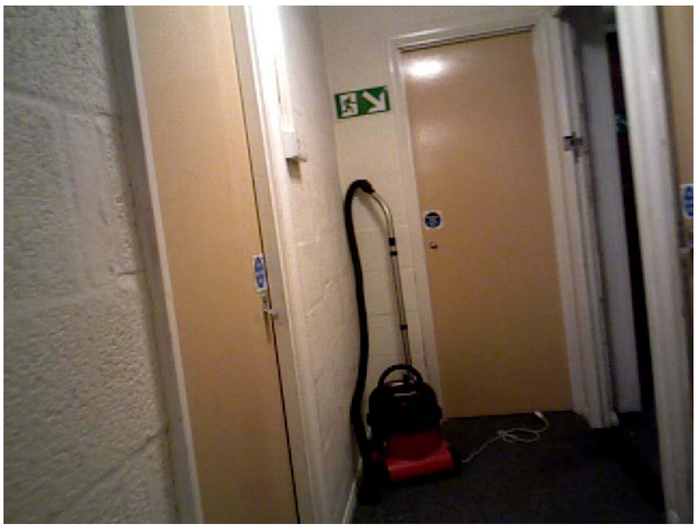
The training is carried out on a desktop computer with an Intel Core i7-3370 @3.4GHz CPU and a GeForce GTX 980 GPU. The software environment is based on TensorFlow [169]. TensorFlow is an open source software library originating from Google's Machine Intelligence research organization for numerical computation using data flow graphs.

The 2D images used for environment construction are directly deployed to train the place recognition CNN. The training dataset contains 20,298 images from 17 locations. Some of the

(a) Elevator with the door closed.


(b) Elevator with the door open and a man walking in.

(c) Storage room with a vacuum cleaner.



(d) Accommodation corridor with a randomly appeared vacuum cleaner.

Figure 5.6: Test images showing the importance of the factor.

training images are shown in Figure 5.7. We have tried three ways to train the Inception-v3 network. Our first attempt is to train the entire network from scratch with random initialization, which is a computationally intensive task. However, we failed to obtain a decent result after training for 3 days since our dataset size is not sufficient enough for the depth of the Inception-v3 network required.

Our second attempt is to use transfer learning strategy to fine-tune a pre-trained model. The pre-trained model is trained on the ImageNet database. We divide our training dataset into training, validation and test subsets based on the ratio 8:2:1. We build the exact same model as Inception-v3 with the number of labels in the final classification layer altered to 17. All weights from the pre-trained model are restored except the final classification layer is randomly initialized.

During this fine-tuning process, all previous weights from all layers can be modified. The smoothed curve in Figure 5.8 evaluates the model precision against training steps. The training time until 20,000 steps is about 8 hours. The precision increases significantly until 14,200 steps and reaches 96.2%, however it starts to drop slightly afterwards. On the other hand, we find the loss generated from the cross-entropy function remains steady after 14,200 steps, as we can see from Figure 5.9. Thus, the precision drop is caused by over-fitting since the model is too complex for our dataset and only particular features in the training images that cannot be applied generally are memorized by the network.

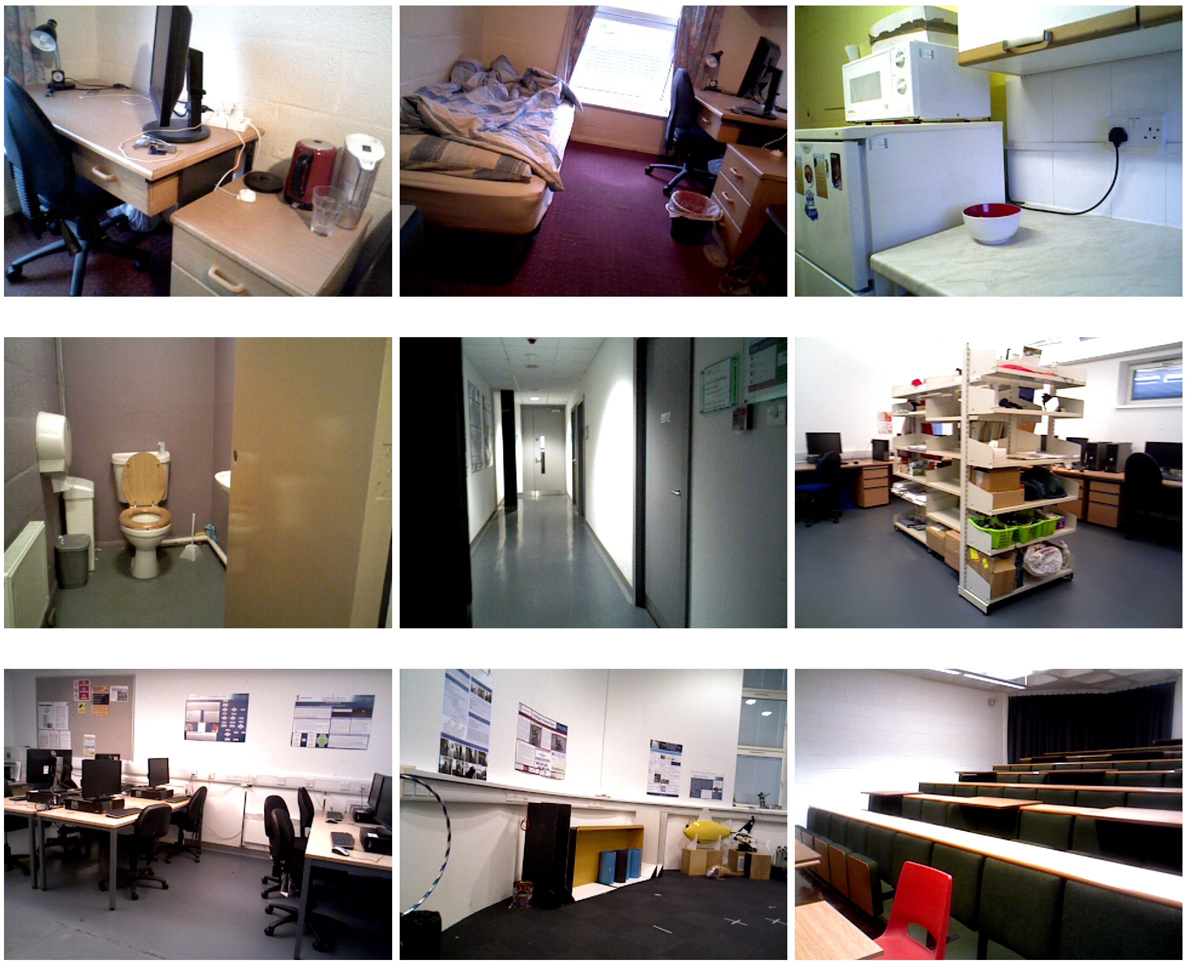


Figure 5.7: Some of the training images at different locations.

In order to reduce training time, the fine-tuning strategy is deployed and the network obtained is saved for further localization in this chapter. More specifically, we only retrain the final classification layer from scratch, while leaving all the rest untouched. In other words, the other layers of the CNN are treated as a fixed feature extractor for our own dataset. This is due to the fact that lower-level portion of a CNN generates more generic features that can be deployed for other tasks, whereas top layer contains relatively more specific features of the dataset used for training. The ratio of the image numbers in training, validation and test subsets is 8:1:1. The initial learning rate is set to a low value so that we obtain a higher overall precision.

We have tested and found that 0.001 gives the best performance. The entire validation subset is used for accuracy calculation to reduce the fluctuation among iterations. However, its drawback is a longer training time. The unsmoothed curve in Figure 5.10 shows the model precision against training steps. The training time until 8,000 steps with an average precision

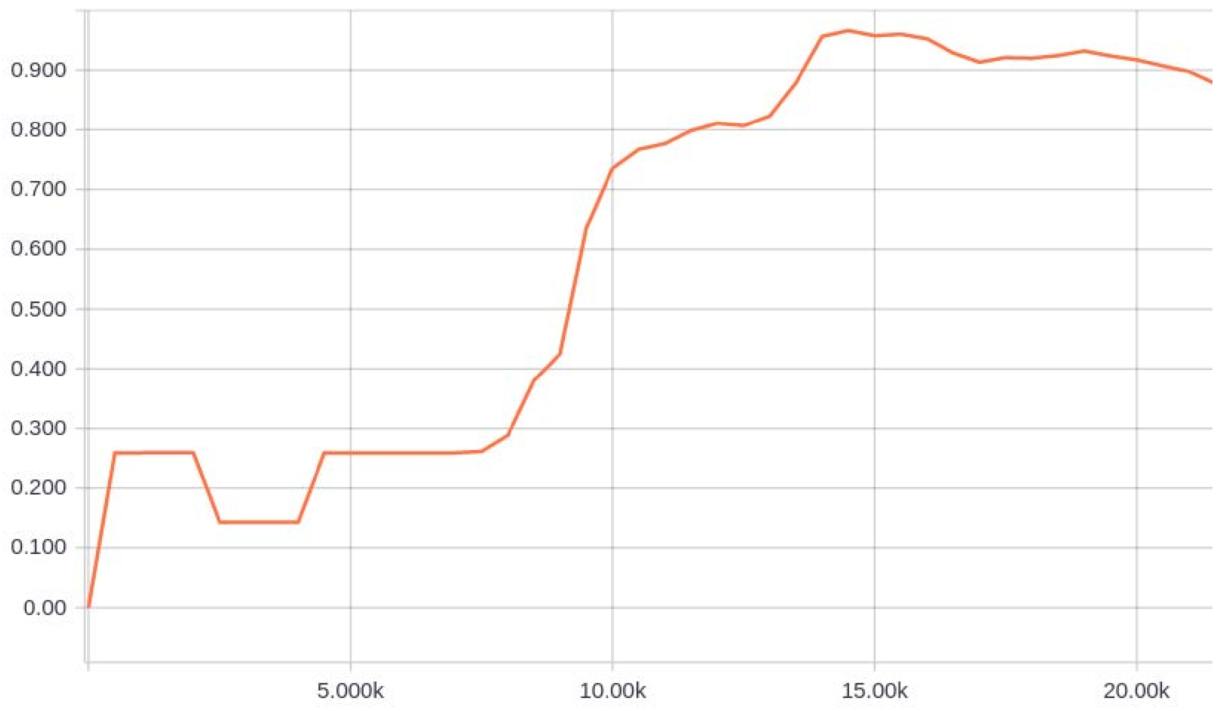


Figure 5.8: Model precision evaluation in the case of fine-tuning among all layers. X-axis: training steps. Y-axis: precision.

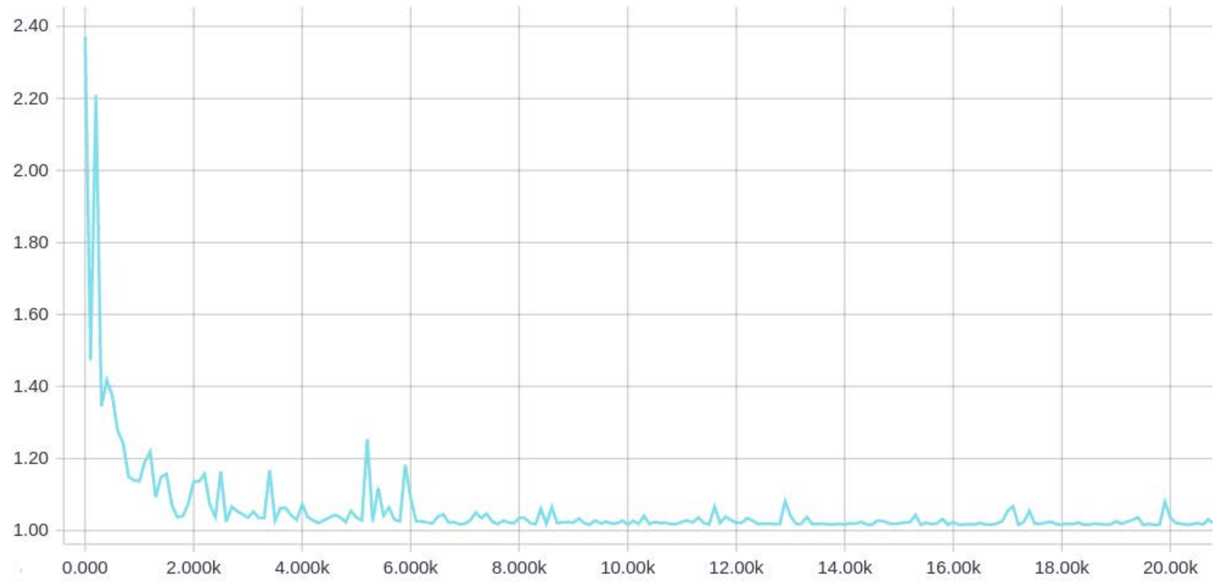


Figure 5.9: Raw cross-entropy loss.

at 97.7% is 24 minutes when a GeForce GTX 980 GPU is used. If only an Intel Core i7-3370 @3.4GHz CPU is used for training, the time is 103 minutes. In this case, the training is much quicker than fine-tuning among all layers and the precision of the trained model is slightly higher.

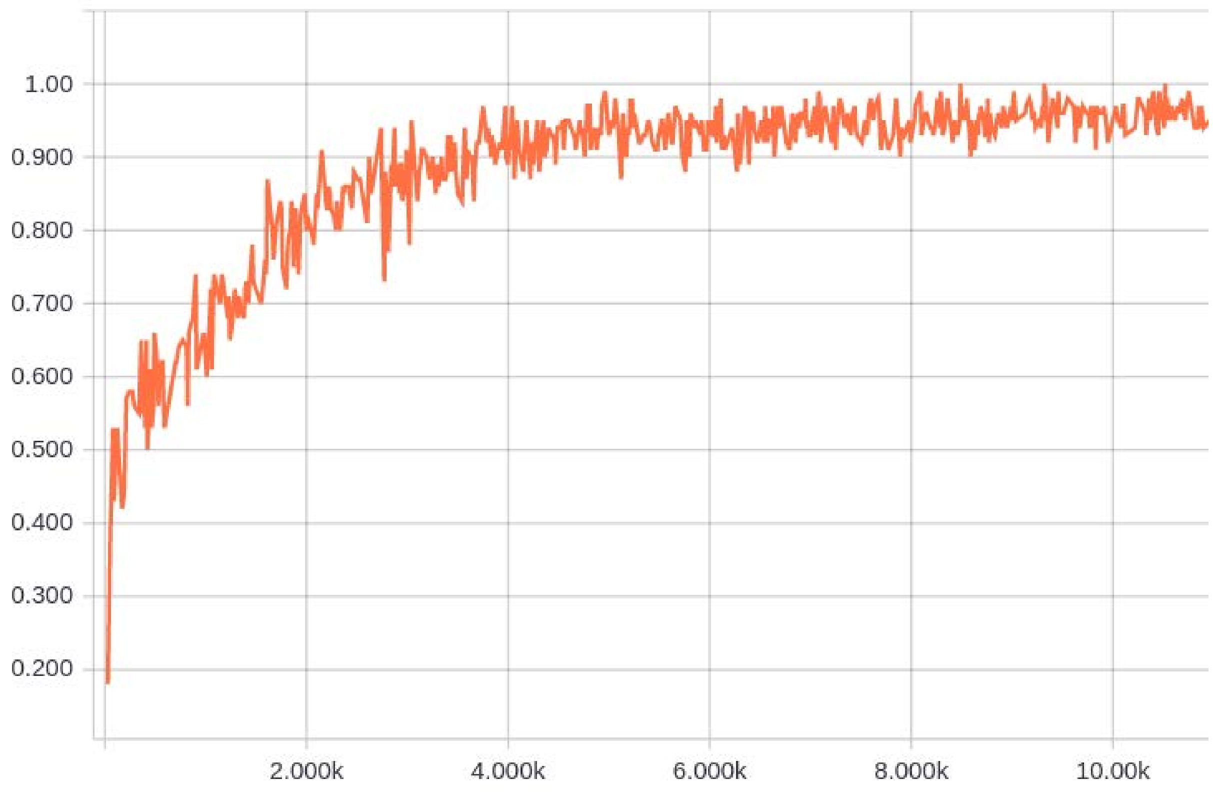


Figure 5.10: Model precision evaluation in the case of fine-tuning only the final layer. X-axis: training steps. Y-axis: precision.

In terms of the CNN for object detection, we adopt the pre-trained Inception-v3 model. Since the model is trained on ImageNet database which has 1,000 labels, we add another linear classifier to minimize the labels. "Bobtail", "chow chow", "tabby cat", etc. are merged into moving animals, "police van", "shark", "military plane" are merged into "others", etc. We have also modified some labels to make them suitable for our task.

5.3.2 Experimental Evaluation

This section describes our experimental results. We compare the performance of our proposed localization system using distinctive objects for further result verification against the end-to-end trained CNN. The home and office environments contain 17 locations in total. We have also considered some locations with similar appearance, such as toilets, corridors, labs and offices.

The objects detected in the mapping process and used for localization are listed in Table 5.1. Some objects are unique objects that can be found at only one place, while others can infer multiple locations.

Table 5.1: Distinctive objects found at each location.

Location	Objects
arena	desk, monitor, tripod, Baxter robot, projector window shade
arena lab	desk, desktop computer, monitor, printer
bedroom	umbrella, running shoe, folding chair, quilt radiator, desk, table lamp, monitor, paper towel backpack, wardrobe, suit
big office	desk, desktop computer, monitor, file cabinet printer
hardware lab	desk, desktop computer, monitor, printer lab chair, oscilloscope
home corridor	corridor
home stairway	banister, handrail
home toilet	washbasin, toilet seat
kitchen	refrigerator, microwave, washbasin, toaster dining table
lecture room	board, desk, folding chair, theater seating
elevator	sliding door
office corridor	sliding door, corridor
office stairway	banister, handrail
office toilet	washbasin, toilet seat
shower room	bathtub, shower curtain, washbasin
small office	desk, desktop computer, monitor, radiator file cabinet, bookcase
storage room	file cabinet, space heater, crutch, mop, desk oscilloscope, croquet ball, project vacuum cleaner, lab chair

We capture new images for testing rather than modifying the images in the training dataset. Since the training images are directly obtained from the mapping process and all scenes have images captured from different viewpoints, the training result indicates the localization accuracy with viewpoint change. Topological localization is similar to place recognition. Thus, precision-recall curves are used to evaluate the performance.

Lighting Condition Change

This section evaluates the influence of lighting condition on the localization performance. The number of test images is 6,875. All objects in the environments remain untouched. Training images are taken in the daytime, while test images are taken at night. For the locations where there are no windows or use window shades all the time, we switch some of the lights off to simulate lighting condition change. Examples are shown in Figure 5.11.

Since the experiment is carried out in indoor environment, the lighting condition has limited impact on both methods. From the precision-recall curves in Figure 5.12, we can see our method performs slightly better than simply using the Inception-v3 model. Based on the entire test dataset, our method results in a 96.3% localization accuracy with the maximum recall rate at 96.0%. Some wrongly identified images are caused by sunlight through windows making shadows on floors and walls.

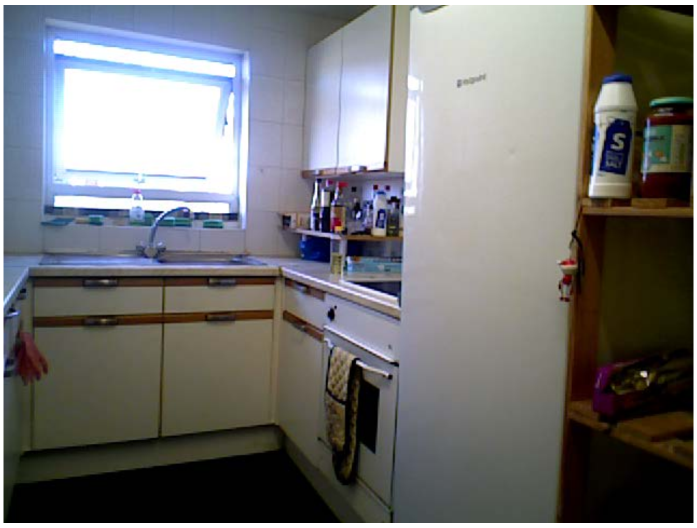
Blurry Images

When a camera is placed on a robot or a wearable device, we cannot guarantee all captured images to be sharp at all times. If the sensor is moving or rotating at a high speed, blurry images are inevitably produced. In this experiment, we test the robustness of our method to these images. There are 2316 blurry images captured in the daytime in this test dataset. Some of them are shown in Figure 5.13. Figure 5.14 shows both methods perform poorly in this experiment. The two curves are almost coincident. The reason is that no object has a higher score than the threshold and then be successful detected in the wrongly identified images except the sliding door of the elevator.

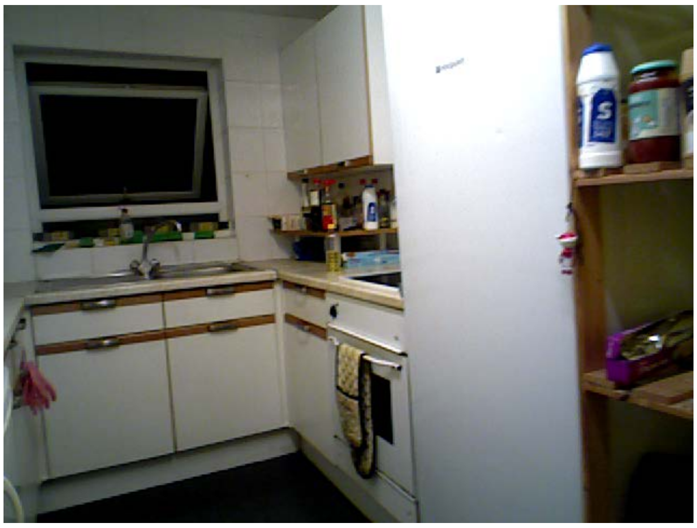
Object Change

One factor that causes obvious appearance change in indoor environment is object change. We leave all the places for common usage and take test images after 1 month. In this experiment, 5,208 images are used for testing and the following conditions are considered.

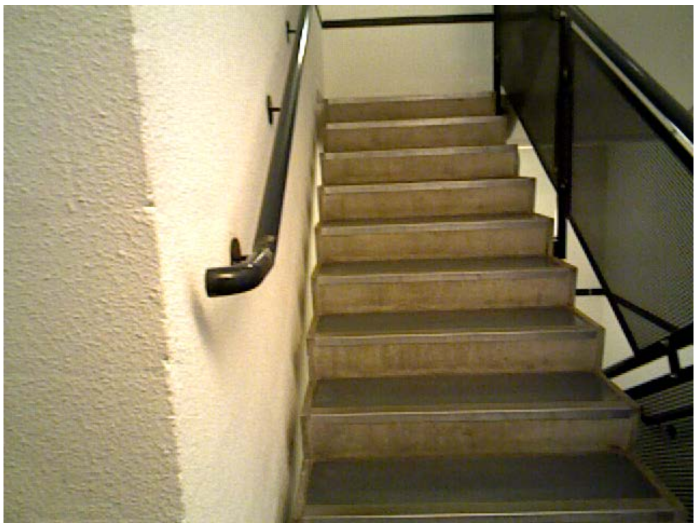
- The locations of objects (chair, kitchen utensil, vacuum cleaner, clothes, elevator door,



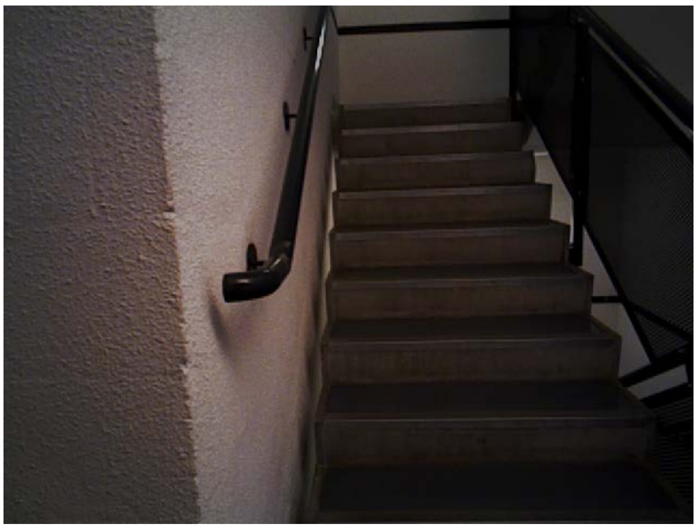
(a) Image taken in the daytime for training.



(b) Image taken in the night for testing.



(c) Image taken with the lights on for training.



(d) Image taken with the lights off for testing.

Figure 5.11: Example images of kitchen with lighting condition change.

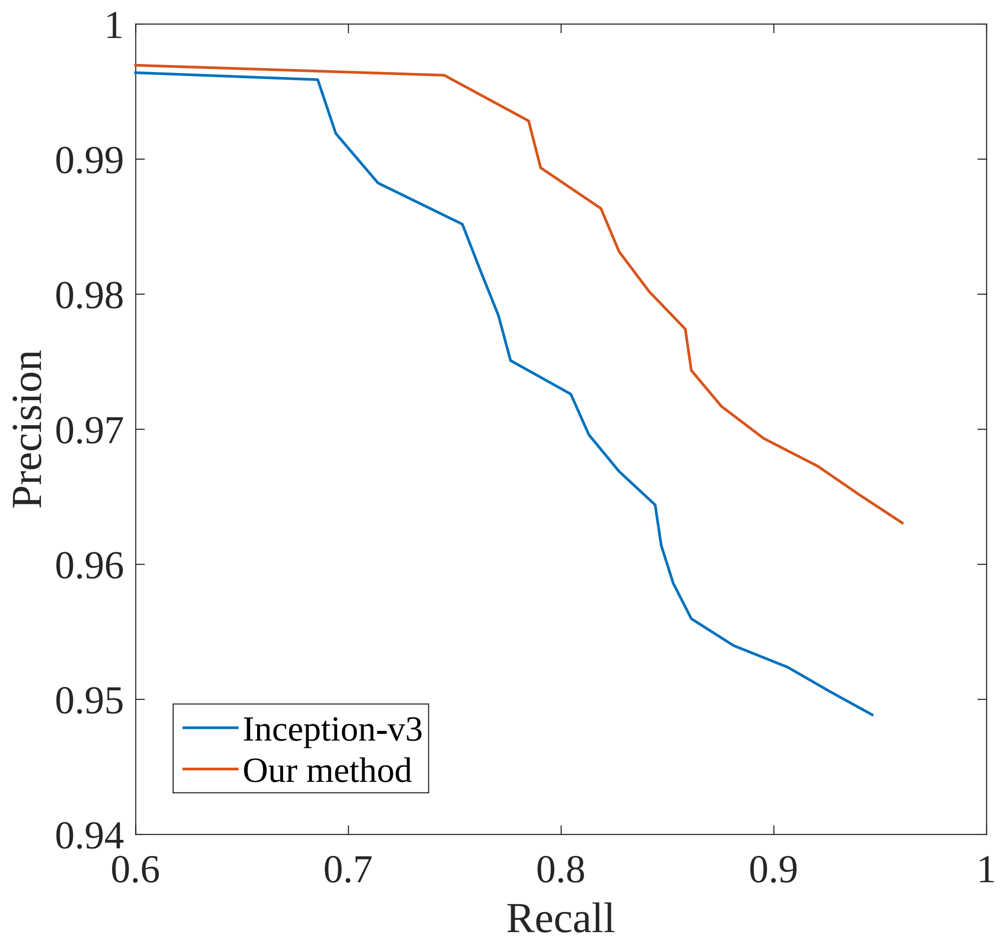
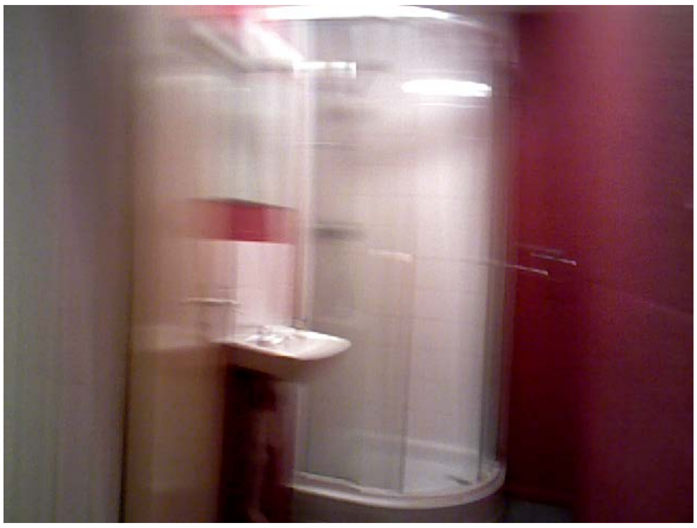
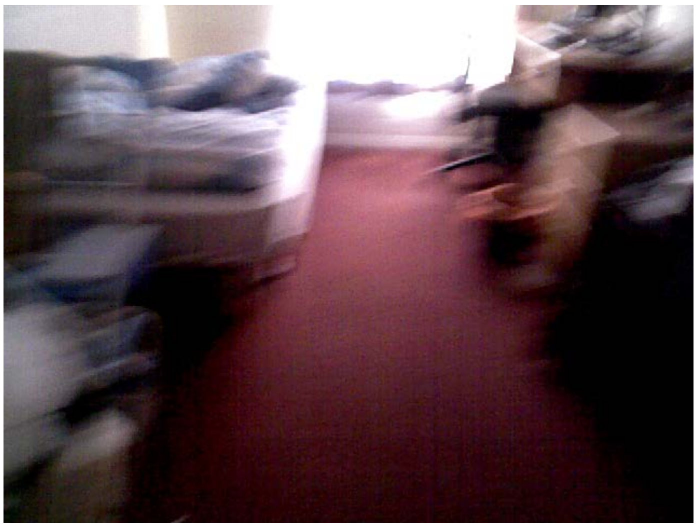
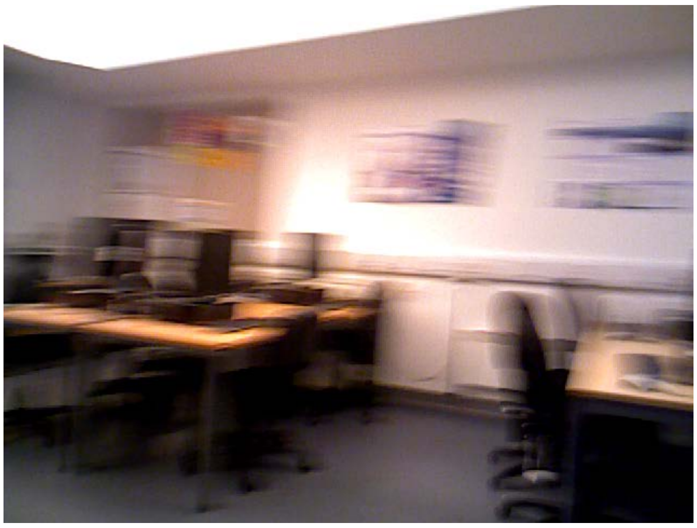
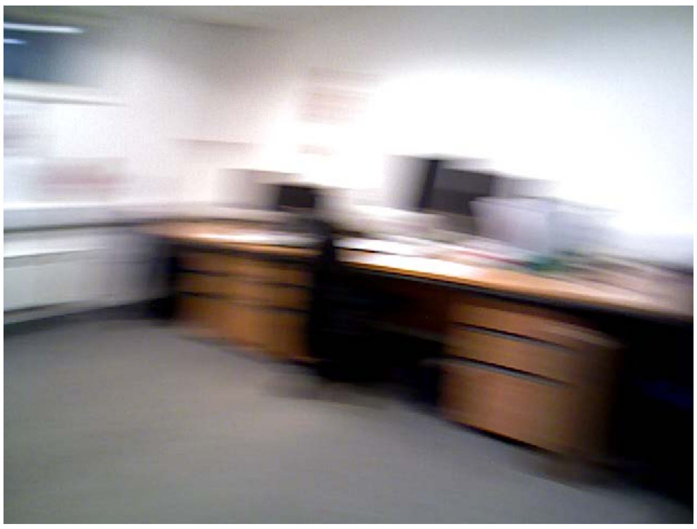


Figure 5.12: Lighting condition change evaluation ($\xi = 1.3$).


(a) Shower room.


(b) Bedroom.


(c) Arena lab.



(d) Small office.

Figure 5.13: Example blurry images for testing.

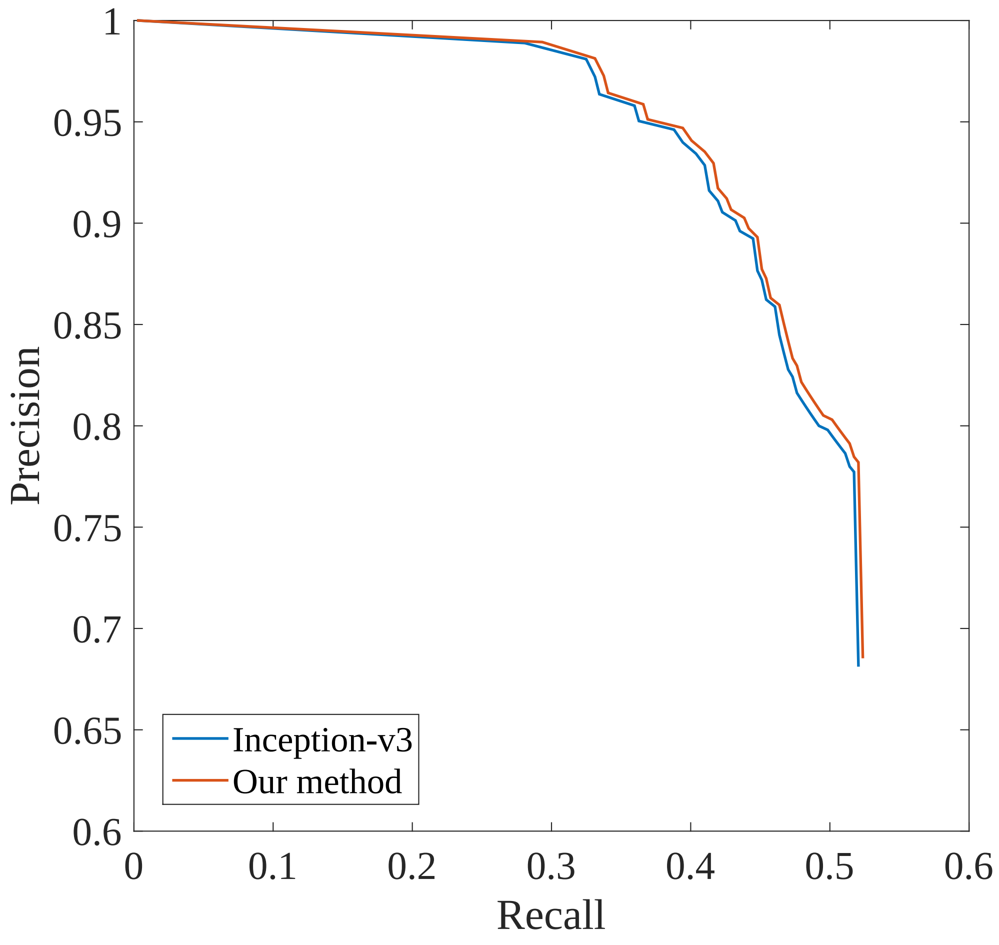
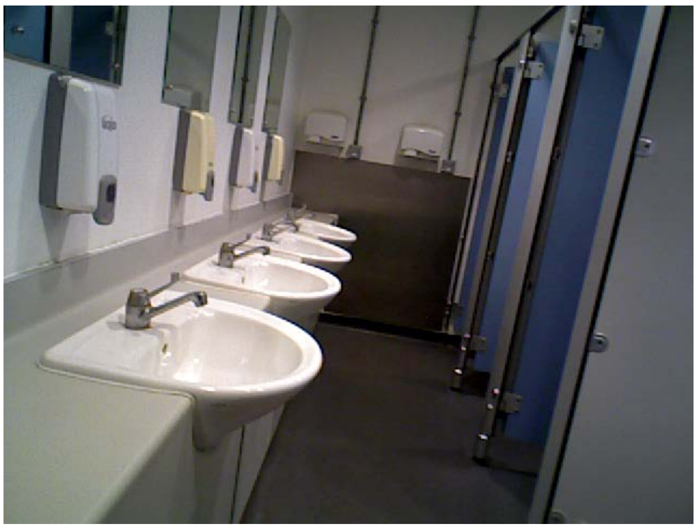


Figure 5.14: Blurry image evaluation ($\xi = 1.3$).


(a) Training image.

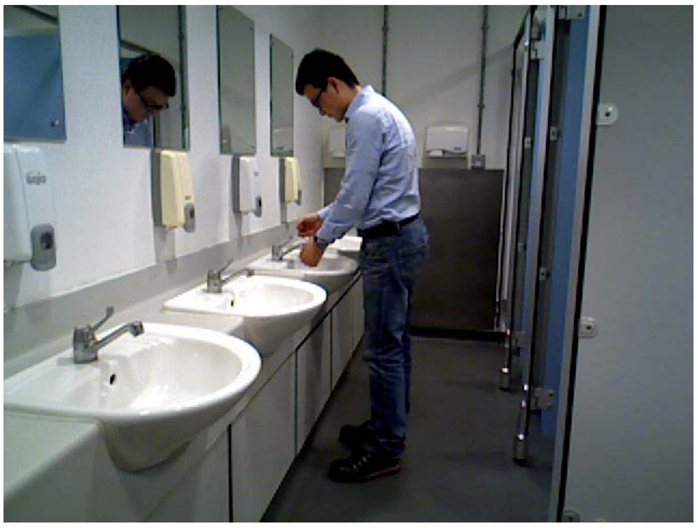

(b) Test image.

Figure 5.15: Example showing how localization results can be verified by detecting objects.

etc.) are changed.

- The deformation of some objects, such as curtain, window shade and quilt.
- New facilities or appliances, such as the stove and oven in the newly refurbished kitchen.
- Randomly appeared humans.

The images in Figure 5.15 gives one example of how the topological localization result is rectified by semantic information in spite of human intervention. The test image wrongly identified as "home corridor" by Inception-v3 is rectified as "office toilet" as the object "washbasin" is detected. Figure 5.16 shows the system performance of these two methods. The precision by using Inception-v3 starts to drop significantly from the recall at 48%, whereas our method starts from 70%. The evaluation on the entire test dataset shows that the precision of using Inception-v3 is 79.9% with the maximum recall at 73.4%, while our method results in a 91.7% precision with the maximum recall at 84.1%.

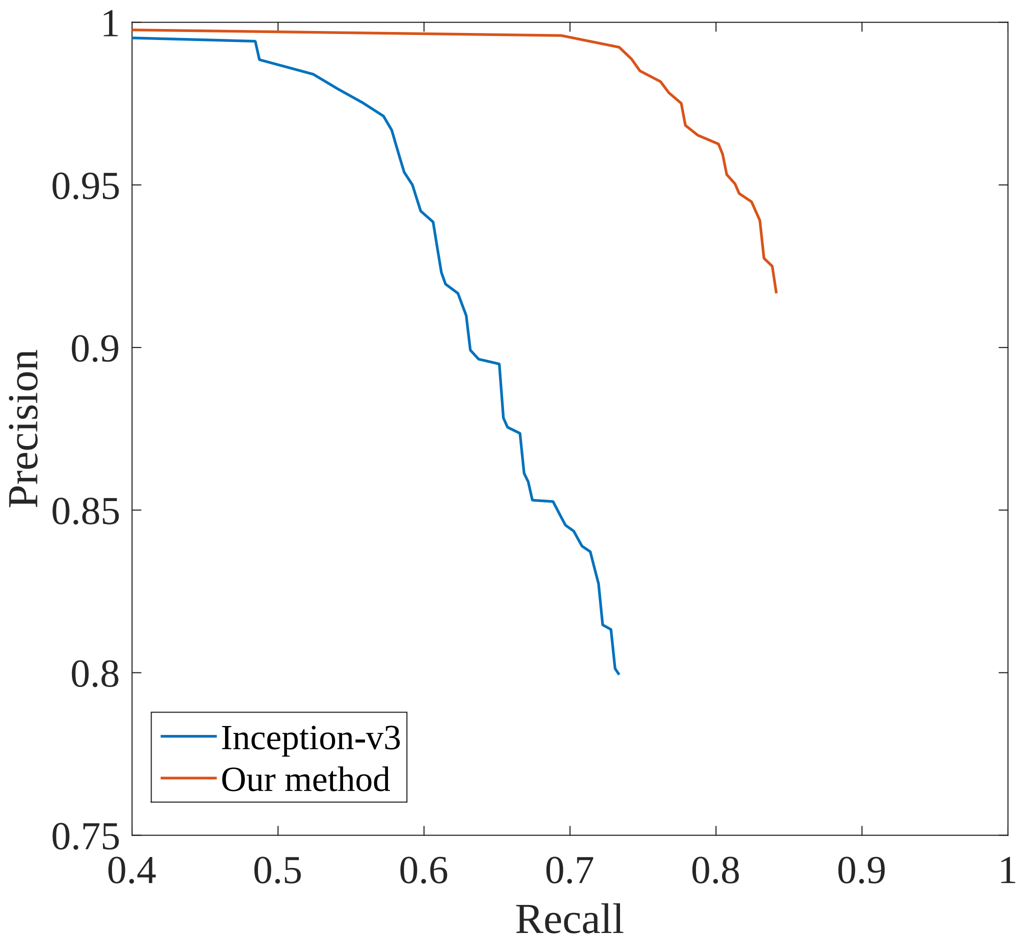


Figure 5.16: Object change evaluation ($\xi = 1.3$).

5.3.3 Factor ξ

The factor ξ plays an important role. Generally speaking, it controls how much the object detection stream is involved in our system. In this section, we evaluate ξ based on the dataset used for object change evaluation. The result is shown in Figure 5.17. We start increasing the value of ξ from 1.1 and use precision-recall curves to test the localization performance. When $\xi = 1.0$, the output scores from the place recognition stream actually remain unchanged. Thus, the curve is same as the one generated by Inception-v3. Both the precision and recall rates increase when we raise the value of ξ. The precision and recall reach the peak at $\xi = 1.3$.

However, if we continue raising the value of ξ, the precision and recall begin to drop. We have also carried out some tests when $\xi > 2$ and found that the curves are all similar to the curve produced by $\xi = 2$. But all of them performs better than the method only using Inception-v3. Therefore, $\xi = 1.3$ is used in all the aforementioned experiments.

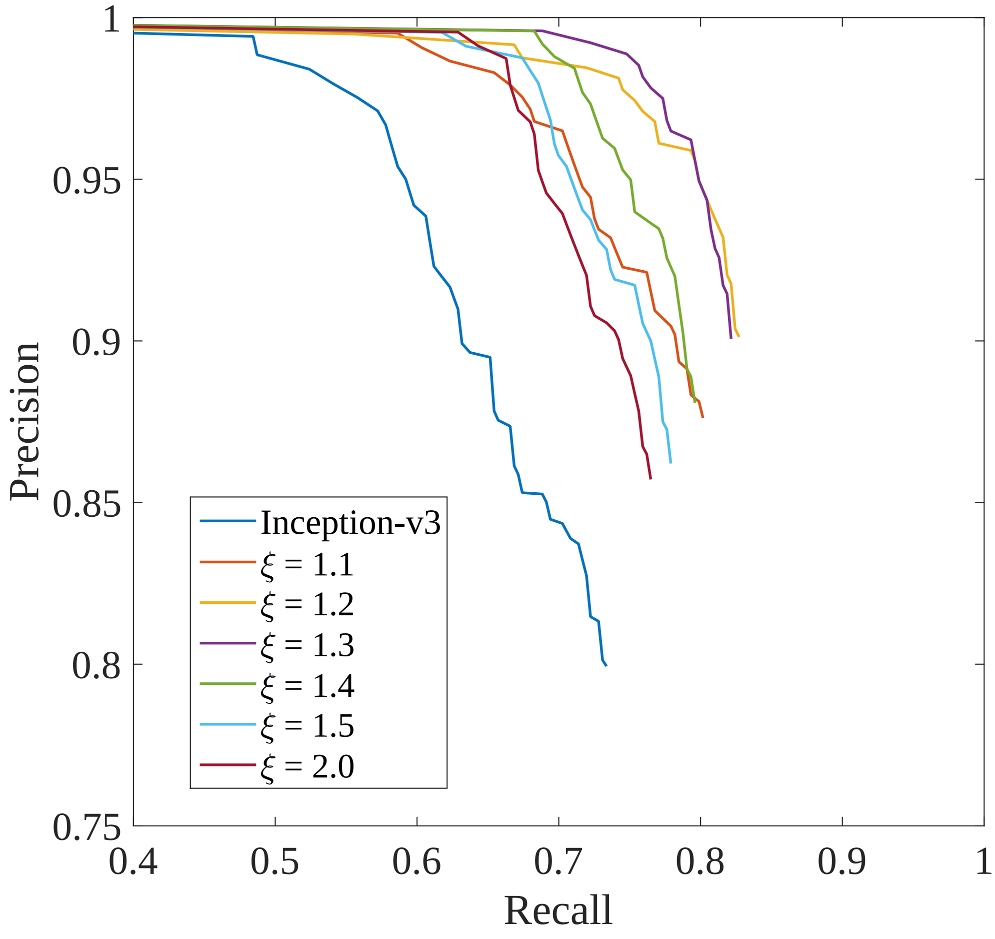


Figure 5.17: Evaluation on factor ξ.

5.3.4 Processing Time

The training time is already detailed in Section 5.3.1. In this section, we used a desktop computer with an Intel Core i7-3370 @3.4GHz CPU, a GeForce GTX 980 GPU and 16GB RAM to test the processing time. The captured image size is 640×480 and then resized to 299×299 pixels. 32 images are placed into one batch. We have also tested the processing time without using the GPU. The test result is presented in Table 5.2. Compared to Inception-v3, our method

Table 5.2: The average processing time of one image.

	Processing Time (unit: second)
Inception-v3 with GPU	0.037
Our method with GPU	0.079
Inception-v3 without GPU	1.492
Our method without GPU	3.432

costs more than twice the processing time.

5.4 Summary

In this part, we address how to use two separately trained CNNs to build a semantic map for vision impaired people and service robots conducting topological indoor localization effectively. The semantic information is used to verify the localization result by detecting distinctive objects within the environment. The performance of our method is analyzed in terms of appearance variation in two indoor environments, such as lighting condition change and object change. Experiments are conducted and the results show that both the precision and recall rates are improved over Inception-v3, apart from a longer processing time. The system can be a wearable device for indoor navigation of visually impaired people or an embedded device for indoor navigation of a mobile robot.

However, the system can not meet real-time requirement without GPUs. In addition, the system takes only one image each time for localization and outputs a result. Thus the relations between frames in a sequence images are abandoned, which are also essential to localization performance.

Chapter 6

Conclusion

This chapter reviews this research, summarizes the achievements made and proposes future work to be conducted.

6.1 Research Summary

Aiming at building a semantic map through low-cost cameras and using the map for autonomous localization, several challenges are addressed and novel approaches are proposed in this thesis. Both semantic mapping and localization methods rely on RGB-D visual data as input. Firstly, a traditional feature based geometrical mapping system is presented and objects in the environments are recognized as semantic information. Secondly, a Recurrent Convolutional Neural Network with an unsupervised end-to-end framework is proposed for camera pose estimation. Lastly, topological localization is carried out when we revisit a scene based on a semantic map built beforehand.

A traditional feature based semantic mapping system is proposed in Chapter 3. The system consists of a 3D camera and a laptop. The metric mapping approach is based on a classic pipeline, namely feature extraction, key-frame detection, camera pose estimation, loop closure detection and global pose graph optimization. Object images need to be stored in the database beforehand. The semantic information extraction approach also relies on SURF feature. The good matching numbers between a key-frame and its neighbouring key-frames are used to eliminate image matching errors. Thus, loop closure detection and object recognition accuracy is increased. We carry out qualitative and quantitative analyses on pose estimation accuracy by using both self-collected dataset and public TUM RGB-D SLAM dataset. Results show the proposed method can be compared to other state-of-the-art algorithms. The performance of semantic information extraction is also verified in a student accommodation environment. No GPU is used in this system to show the calculation efficiency.

A novel deep learning based camera pose estimation approach is presented in Chapter 4. The system adopts an unsupervised end-to-end training approach. A VGG based Convolutional Neural Network is designed as a feature extractor and a LSTM network is used as a pose estimator. In the case of training, we feed pairs of consecutive monocular and depth images into the RCNN. No ground truth pose labeling is needed, thus the proposed system requires less human effort and is cheap to run. Only monocular images are used for testing and the estimated camera poses are on a absolute scale. Qualitative and quantitative analyses are carried out based on one of the most popular VO dataset, KITTI. Results have shown that our system can be compared to stereo VO systems and outperforms other monocular VO systems in terms of both translation and rotation accuracy. The pre-trained model can also be generalized to other similar scenes and the performance of the system can be improved by further training with larger dataset. Compared to feature based systems, this system requires GPUs to run, thus is computationally expensive.

A novel topological localization approach based on pre-built semantic maps is detailed in Chapter 5. A CNN is used for object recognition. Only object names are required in the database rather than object images, thus making semantic information extraction more flexible. We use two steams of CNNs for topological localization. The system takes one RGB image and directly outputs the inferred locations. The extracted semantic information is inversely used to verify localization results. Experiments are conducted in terms of appearance variation such as viewpoint, lighting condition and object changes. Precision-recall curves are leveraged to evaluate the proposed approach. Results show that the additional semantic information can boost localization performance, especially when significant appearance change exists.

6.2 Thesis Contributions

This thesis research has made a number of contributions listed below:

(1) **Feature based semantic mapping**

We propose a feature based semantic mapping approach using an RGB-D sensor. Camera poses are estimated through both RGB and depth images, whereas semantic information is extracted from only RGB images. SURF feature is adopted for both tasks. We describe images with descriptors and use the relations between two consecutive frames to estimate a local transformation matrix. If the transformation is substantial enough, a new key-frame is added. Loop closure detection is subsequently carried out by matching a key-frame to some of the previous key-frames. The global pose graph can thus be constructed by connecting sequential key-frames and is then optimized by g2o. Finally, we down-sample each key-frame and project pairs of RGB and depth images into a common coordinate frame. As for semantic information extraction, we take some pictures of the objects in the environment, associate them with natural language and store them in a database beforehand. We then apply object recognition in each of the key-frames along with geometrical reconstruction. A novel approach is presented to diagnose and eliminate errors during semantic extraction. The approach can also be applied to loop closure detection. The global semantic map can then be stored and used for further inferring and navigation.

(2) **Visual odometry based on unsupervised deep learning**

We present a novel monocular visual odometry system based on an unsupervised Recurrent Convolutional Neural Network. The system benefits from an unsupervised end-to-end framework, thus no ground truth camera poses are required for training. Instead, the ground truth poses of the camera are only used for performance evaluation. Therefore, such unsupervised training eliminates the need of the labour-intensive image labeling task. In addition, the performance of our system can be easily improved by further training with larger unlabeled dataset. Both CNN and RNN are leveraged for this task. The CNN, being a feed-forward network, learns to differentiate patterns across space, thus can be regarded as a feature extractor. The RNN learns to recognize patterns across time and can be viewed as a pose estimator. The total loss consists of 2D and 3D spacial losses. On the other hand, absolute scale can be recovered without pose post-processing. To inject scale, depth information of scenes obtained by a 3D LiDAR is used alongside monocular images to train the network. Poses are inferred only from monocular images, thus making the proposed visual odometry system a monocular one. Experiments have been carried out on KITTI odometry dataset and results have shown that the proposed VO system can be compared to other state-of-the-art monocular VO systems in terms of both translation and rotation accuracy even without scale post-processing.

(3) **Topological localization based on deep learning**

We present a novel localization approach to support the indoor localization of people with vision impairment. A 3D indoor semantic map is firstly constructed using an RGB-D sensor. To inject semantic information, we adopt deep Convolutional Neural Networks for daily object detection, rather than the bag-of-visual-words model (BoW). This enables the objects to be detected in a much broader scope. Objects in any shape, form or colour can be classified into one category as long as they have the same name. Thus, only object names are stored in our database rather than images, which results in a relatively small database. Once objects are detected, the subordination of the objects to rooms is then represented by the anchoring method. The constructed map is then deployed to help visually impairment users conducting indoor topological localization. Distinctive objects detected and labeled from the mapping process, as well as the undetected objects that might exist and can infer the function of a room such as “bedroom” or “lab”, are used to further verify locations. The semantic information can thus be used to diagnose and eliminate errors and boost topological localization performance despite appearance changes within the environment. For instance, the lighting conditions vary between day and night, and objects (chair, laptop, curtain or even human) may be in random shapes or locations. Experiments with long-term operations have also been carried out and shown how semantic information accounts for appearance change within the environment. Compared to other state-of-the-art algorithms, our method generates higher precision and recall rates.

6.3 A List of Publications

The academic publications achieved during this PhD study are listed as follows.

(1) Qiang Liu, Ruihao Li, Huosheng Hu and Dongbing Gu. Using Unsupervised Deep Learning Technique for Monocular Visual Odometry. IEEE Transactions on Cybernetics, 2018. (Under Review)

(2) Qiang Liu, Ruihao Li, Huosheng Hu and Dongbing Gu. Using Deep Learning Technique to Build a Semantic Map for Topological Localization. Cognitive Computation, 2017. (Under Review)

(3) Qiang Liu, Ruihao Li, Huosheng Hu and Dongbing Gu. Extracting Semantic Information from Visual Data: A Survey. Robotics 5. no. 1 (2016): 8.

(4) Qiang Liu, Ruihao Li, Huosheng Hu and Dongbing Gu. Using Semantic Maps for Room Recognition to Aid Visually Impaired People. In the 22th International Conference on Automation and Computing (ICAC), pp. 89-94. IEEE, 2016.

(5) Qiang Liu, Ruihao Li, Huosheng Hu and Dongbing Gu. Building Semantic Maps for Blind People to Navigate at Home. In the 8th Computer Science and Electronic Engineering Conference (CEEC), pp. 12-17. IEEE, 2016.

(6) Ruihao Li, Qiang Liu, Jianjun Gui, Huosheng Hu, Dongbing Gu. Indoor Relocalization in Challenging Environments with Dual-stream Convolutional Neural Networks. IEEE Transactions on Automation Science and Engineering (T-ASE), 2017.

(7) Ruihao Li, Dongbing Gu, Qiang Liu, Zhiqiang Long, Huosheng Hu. Semantic Scene Mapping with Spatial-temporal Deep Neural Network for Robotic Applications. Cognitive Computation 10, no. 2 (2018): 260-271.

(8) Ruihao Li, Qiang Liu, Jianjun Gui, Huosheng Hu, Dongbing Gu. A Novel RGB-D SLAM Algorithm Based on Points and Plane-Patches. In IEEE International Conference on Automation Science and Engineering (CASE), pp. 1348-1353. IEEE, 2016.

(9) Ruihao Li, Qiang Liu, Jianjun Gui, Huosheng Hu, Dongbing Gu. Night-time Indoor Relocalization Using Depth Image with Convolutional Neural Network. In the 22th International Conference on Automation and Computing (ICAC), pp. 261-266. IEEE, 2016.

6.4 Future Work

This thesis presents novel semantic mapping and localization methods based on 3D visual data. Some challenges have be addressed, yet several problems remain and can be summarized as follows.

- The resolution of the images used for training deep learning networks is important. The blurry images produced by low accuracy and resolution cameras are hard to use by a robot to recognize objects in the real-world. Apart from the improvement of hardware, several software methodologies such as super-resolution or data fusion could be deployed for accurate semantic mapping. Super-resolution is a class of techniques that enhance the resolution of an imaging system. Data fusion is a process of integrating multiple data and knowledge representing the same scene in order to produce an accurate output.

- The current feature extractors need to be extended. Although features extracted by deep learning techniques are robust to geometric transformations and illumination changes, the features are quite limited to the appearance of objects such as edges, corners and their relations. Extracting the semantic inherent characteristic of objects might be a solution, e.g. the legs of a chair, the keyboard and display of a laptop, etc.
- Classifiers should be adaptive to the dynamic changes in the real-world. The current semantic mapping systems need pre-training, and can only recognize the trained objects or certain scenes. However, the real-world environments are changing dynamically, and object appearances are changing all the time. Any semantic mapping algorithms need the ability of self-learning to adapt these changes and recognize new objects. Solutions might be found in the deep learning domain.
- Depth data can be incorporated during geometrical or topological localization in order to improve system performance for real-time navigation of autonomous robots and the visual guidance of blind people. Compared to conventional RGB images, depth data also has rich information and works well in certain conditions such as in the night or in bright sunlight. Moreover, the combination of monocular images and depth information is widely applied recently such as self-driving cars and smart phones, which might be a standard configuration of the electronic products in the future.
- Semantic mapping systems should be able to detect novelty and learn novel concepts about the environment continuously and in real time. The conceptual definitions that are initially encoded by using common sense knowledge should be updated or extended based on new experience. For instance, a robot operating in a home environment should link its actions to rooms and objects in order to bridge the gap between metric map and semantic knowledge. Moreover, this conceptual learning performance opens new possibilities in terms of truly autonomous semantic mapping and navigation. Thus, an incremental adaptive learning or active learning model should be built.

Bibliography

[1] Mapping (ROS). [Online]. Available: http://rrt.fh-wels.at/sites/robocup/mapping.html

[2] A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat, "Incremental vision-based topological SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 1031–1036.

[3] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt, "Multi-modal semantic place classification," *The International Journal of Robotics Research (IJRR)*, vol. 29, pp. 298–320, 2009.

[4] N. Blodow, L. C. Goron, Z.-C. Marton, D. Pangercic, T. Rühr, M. Tenorth, and M. Beetz, "Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4263–4270.

[5] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[6] P. Cavalcante, M. Sehili, M. Herbin, D. Istrate, F. Blanchard, J. Boudy, and B. Dorizzi, "First steps in adaptation of an evidential network for data fusion in the framework of medical remote monitoring," in *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2012, pp. 2044–2047.

[7] SYSIASS: SYSteme Intelligent et Autonome d'aide aux Soins de Sante / Autonomous and Intelligent Healthcare System. [Online]. Available: http://www.sysiass.eu/

[8] COALAS: Cognitive Assisted Living Ambient System. [Online]. Available: http://coalas-project.eu/

[9] K. Severinson-Eklundh, A. Green, and H. Hüttenrauch, “Social and collaborative aspects of interaction with a service robot,” *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 223–234, 2003.

[10] S. Park and S. Jayaraman, “Enhancing the quality of life through wearable technology,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 3, pp. 41–48, 2003.

[11] O. Atzmon, “Innovative mobile-health solutions promise to improve elderly care and save costs,” *MST NEWS*, vol. 5, p. 42, 2005.

[12] N. Noury, P. Rumeau, A. Bourke, G. ÓLaighin, and J. Lundy, “A proposal for the classification and evaluation of fall detectors,” *IRBM*, vol. 29, no. 6, pp. 340–349, 2008.

[13] D. Dakopoulos and N. G. Bourbakis, “Wearable obstacle avoidance electronic travel aids for blind: a survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 25–35, 2010.

[14] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: Part I,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[15] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[16] I. Arel, D. C. Rose, T. P. Karnowski *et al.*, “Deep machine learning-a new frontier in artificial intelligence research,” *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.

[17] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, “Visual modeling with a hand-held camera,” *International Journal of Computer Vision*, vol. 59, no. 3, pp. 207–232, 2004.

[18] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2006, pp. 2169–2178.

[19] T. Fong, I. Nourbakhsh, and K. Dautenhahn, “A survey of socially interactive robots,” *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 143–166, 2003.

[20] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1052–1067, 2007.

[21] Q. Liu, R. Li, H. Hu, and D. Gu, "Extracting semantic information from visual data: A survey," *Robotics*, vol. 5, no. 1, p. 8, 2016.

[22] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2043–2050.

[23] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1027–1037, 2008.

[24] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 2002, pp. 1014–1019.

[25] J. Wu, H. Christensen, J. M. Rehg *et al.*, "Visual place categorization: Problem, dataset, and algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 4763–4770.

[26] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3515–3522.

[27] S. Thrun *et al.*, "Robotic mapping: A survey," *Exploring artificial intelligence in the new millennium*, pp. 1–35, 2002.

[28] D. F. Wolf and G. S. Sukhatme, "Semantic mapping using mobile robots," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 245–258, 2008.

[29] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots - an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359–371, 2007.

[30] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.

[31] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[32] V. Harmandas, M. Sanderson, and M. Dunlop, "Image retrieval by hypertext links," in *The 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 1997, pp. 296–303.

[33] H. Zhuge, "Retrieve images by understanding semantic links and clustering image fragments," *Journal of Systems and Software*, vol. 73, no. 3, pp. 455–466, 2004.

[34] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[35] C. Theobalt, J. Bos, T. Chapman, A. Espinosa-Romero, M. Fraser, G. Hayes, E. Klein, T. Oka, and R. Reeve, "Talking to Godot: Dialogue with a mobile robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, 2002, pp. 1338–1343.

[36] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 2, pp. 154–167, 2004.

[37] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernández-Madrigal, J. Gonzalez *et al.*, "Multi-hierarchical semantic maps for mobile robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 2278–2283.

[38] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla, "Evaluation of low-complexity visual feature detectors and descriptors," in *IEEE International Conference on Digital Signal Processing (DSP)*, 2013, pp. 1–7.

[39] C. Siagian and L. Itti, "Biologically inspired mobile robot vision localization," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 861–873, 2009.

[40] D. Lisin, M. Mattar, M. B. Blaschko, E. G. Learned-Miller, M. C. Benfield *et al.*, "Combining local and global image features for object class recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 47–47.

[41] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *IEEE International Conference on Computer Vision (ICCV)*, 1998, pp. 555–562.

[42] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision (IJCV)*, vol. 7, no. 1, pp. 11–32, 1991.

[43] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[44] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen, "A discriminative approach to robust visual place recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 3829–3836.

[45] K. Murphy, A. Torralba, W. Freeman *et al.*, "Using the forest to see the trees: A graphical model relating features, objects and scenes," *Advances in Neural Information Processing Systems (NIPS)*, vol. 16, pp. 1499–1506, 2003.

[46] O. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.

[47] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2000, pp. 1023–1029.

[48] D. Filliat, E. Battesti, S. Bazeille, G. Duceux, A. Gepperth, L. Harrath, I. Jebari, R. Pereira, A. Tapus, C. Meyer *et al.*, "RGBD object recognition and visual texture classification for indoor semantic mapping," in *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 2012, pp. 127–132.

[49] H. Grimmett, M. Buerki, L. Paz, P. Pinies, P. Furgale, I. Posner, and P. Newman, "Integrating metric and semantic maps for vision-only automated parking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2159–2166.

[50] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 300–312, 2007.

[51] K. Li and M. Q.-H. Meng, "Indoor scene recognition via probabilistic semantic map," in *IEEE International Conference on Automation and Logistics (ICAL)*, 2012, pp. 352–357.

[52] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 858–865.

[53] Y. Li, S. Wang, Q. Tian, and X. Ding, "A survey of recent advances in visual feature detection," *Neurocomputing*, vol. 149, pp. 736–751, 2015.

[54] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[55] C. Harris and M. Stephens, "A combined corner and edge detector," in *The 4th Alvey Vision Conference*, vol. 15. Citeseer, 1988, p. 50.

[56] C. Tomasi, "Shape and motion from image streams: a factorization method part 3: Detection and tracking of point features," *Tech. Report*, 1991.

[57] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision (ECCV)*. Springer, 2006, pp. 430–443.

[58] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 183–196.

[59] M. Agrawal, K. Konolige, and M. R. Blas, “CenSurE: Center surround extremas for realtime feature detection and matching,” in *European Conference on Computer Vision (ECCV)*. Springer, 2008, pp. 102–115.

[60] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[61] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[62] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *European Conference on Computer Vision (ECCV)*. Springer, 2006, pp. 404–417.

[63] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” in *European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 778–792.

[64] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[65] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary robust invariant scalable keypoints,” in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2548–2555.

[66] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast retina keypoint,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 510–517.

[67] M. Basu, “Gaussian-based edge-detection methods - a survey,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 32, no. 3, pp. 252–260, 2002.

[68] A. Ranganathan and F. Dellaert, “Semantic modeling of places using objects,” in *Robotics: Science and Systems Conference*, vol. 3, 2007, pp. 27–30.

[69] S. Kim and M.-S. Shim, “Biologically motivated novel localization paradigm by high-level multiple object recognition in panoramic images,” *The Scientific World Journal*, vol. 2015, p. 465290, 2015.

[70] G. Tsai and B. Kuipers, “Dynamic visual understanding of the local environment for an indoor navigating robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4695–4701.

[71] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments,” *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 5, pp. 647–663, 2012.

[72] D. Gálvez-López and J. D. Tardós, “Bags of binary words for fast place recognition in image sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[73] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[74] D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe, “Curious George: An attentive semantic robot,” *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 503–511, 2008.

[75] H. Zender, O. M. Mozos, P. Jensfelt, G.-J. Kruijff, and W. Burgard, “Conceptual spatial representations for indoor mobile robots,” *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, 2008.

[76] I. Kostavelis, K. Charalampous, A. Gasteratos, and J. K. Tsotsos, “Robot navigation via spatial and temporal coherent semantic maps,” *Engineering Applications of Artificial Intelligence*, vol. 48, pp. 173–187, 2016.

[77] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[78] L. Riazuelo, M. Tenorth, D. Di Marco, M. Salas, D. Gálvez-López, L. Mosenlechner, L. Kunze, M. Beetz, J. D. Tardos, L. Montano *et al.*, “RoboEarth semantic mapping: A cloud enabled knowledge-based approach,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 432–443, 2015.

[79] B. Morisset, R. B. Rusu, A. Sundaresan, K. Hauser, M. Agrawal, J.-C. Latombe, and M. Beetz, “Leaving flatland: Toward real-time 3D navigation,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 3786–3793.

[80] I. Kostavelis and A. Gasteratos, “Learning spatially semantic representations for cognitive robot navigation,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1460–1475, 2013.

[81] M. Waibel, M. Beetz, J. Civera, R. dâĂŹAndrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. Montiel, A. Perzylo *et al.*, “A world wide web for robots,” *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, 2011.

[82] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, “A real-time RGB-D registration and mapping approach by heuristically switching between photometric and geometric information,” in *The 17th International Conference on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–8.

[83] L. Zhang, K. Mistry, M. Jiang, S. C. Neoh, and M. A. Hossain, “Adaptive facial point detection and emotion recognition for a humanoid robot,” *Computer Vision and Image Understanding*, vol. 140, pp. 93–114, 2015.

[84] H. Sun, C. Wang, and N. El-Sheimy, “Automatic traffic lane detection for mobile mapping systems,” in *International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM)*. IEEE, 2011, pp. 1–5.

[85] C. Schmid, R. Mohr, and C. Bauckhage, “Evaluation of interest point detectors,” *International Journal of Computer Vision (IJCV)*, vol. 37, no. 2, pp. 151–172, 2000.

[86] R. F. Salas-Moreno, “Dense semantic SLAM,” Ph.D. dissertation, Imperial College London, UK, 2014.

[87] A. Torralba, K. P. Murphy, W. T. Freeman, M. Rubin *et al.*, "Context-based vision system for place and object recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 273–280.

[88] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," *Pattern Recognition*, vol. 2781, pp. 297–304, 2003.

[89] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[90] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, "Spectral partitioning with indefinite kernels using the Nyström extension," in *European Conference on Computer Vision (ECCV)*. Springer, 2002, pp. 531–542.

[91] S. Taylor and T. Drummond, "Multiple target localisation at over 100 FPS," in *British Machine Vision Conference (BMVC)*, 2009, pp. 1–11.

[92] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant Orientation Templates for real-time detection of texture-less objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2257–2264.

[93] R. Anati, D. Scaramuzza, K. G. Derpanis, and K. Daniilidis, "Robot localization using soft object detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 4992–4999.

[94] J. Stückler, N. Biresev, and S. Behnke, "Semantic mapping using object-class segmentation of RGB-D images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 3005–3010.

[95] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *IEEE International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[96] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3921–3926.

[97] J. Martínez-Gómez, V. Morell, M. Cazorla, and I. García-Varea, "Semantic localization in the pcl library," *Robotics and Autonomous Systems*, vol. 75, pp. 641–648, 2016.

[98] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 1470–1477.

[99] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision (IJCV)*, vol. 43, no. 1, pp. 29–44, 2001.

[100] C. Case, B. Suresh, A. Coates, and A. Y. Ng, "Autonomous sign reading for semantic mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3297–3303.

[101] M. Sami, Y. Ayaz, M. Jamil, S. O. Gilani, and M. Naveed, "Text detection and recognition for semantic mapping in indoor navigation," in *The 5th International Conference on IT Convergence and Security (ICITCS)*, 2015, pp. 1–4.

[102] M. Wyss and P. I. Corke, "Active text perception for mobile robots," 2012.

[103] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[104] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[105] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using Convolutional Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1717–1724.

[106] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[107] N. Sunderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, “On the performance of ConvNet features for place recognition,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4297–4304.

[108] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[109] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A convolutional network for real-time 6-DOF camera relocalization,” in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2938–2946.

[110] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich *et al.*, “Going deeper with convolutions,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[111] R. Li, Q. Liu, J. Gui, D. Gu, and H. Hu, “Indoor relocalization in challenging environments with dual-stream convolutional neural networks,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 651–662, 2018.

[112] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2625–2634.

[113] S. Wang, R. Clark, H. Wen, and N. Trigoni, “End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks,” *The International Journal of Robotics Research (IJRR)*, vol. 37, no. 4-5, pp. 513–542, 2018.

[114] C. Galindo, J.-A. Fernández-Madrigal, J. González, and A. Saffiotti, “Robot task planning using semantic maps,” *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 955–966, 2008.

[115] F. Baader, *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[116] S. Coradeschi and A. Saffiotti, “An introduction to the anchoring problem,” *Robotics and Autonomous Systems*, vol. 43, no. 2, pp. 85–96, 2003.

[117] S. Vasudevan and R. Siegwart, “Bayesian space conceptualization and place classification for semantic maps in mobile robotics,” *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 522–537, 2008.

[118] P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. Mackworth, “Automated spatial-semantic modeling with applications to place labeling and informed search,” in *The 6th Canadian Conference on Computer and Robot Vision (CRV)*, 2009, pp. 284–291.

[119] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “LabelMe: A database and web-based tool for image annotation,” *International Journal of Computer Vision (IJCV)*, vol. 77, no. 1-3, pp. 157–173, 2008.

[120] R. Capobianco, J. Serafin, J. Dichtl, G. Grisetti, L. Iocchi, and D. Nardi, “A proposal for semantic map representation and evaluation,” in *European Conference on Mobile Robots (ECMR)*, 2015, pp. 1–6.

[121] C. Galindo, J. González, J.-A. Fernández-Madrigal, and A. Saffiotti, “Robots that change their world: Inferring goals from semantic knowledge,” in *European Conference on Mobile Robots (ECMR)*, 2011, pp. 1–6.

[122] J. Crespo Herrero, R. I. Barber Castano, and O. Martinez Mozos, “An inferring semantic system based on relational models for mobile robotics,” in *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2015, pp. 83–88.

[123] D. W. Ko, C. Yi, and I. H. Suh, “Semantic mapping and navigation: A Bayesian approach,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 2630–2636.

[124] A. Boularias, F. Duvallet, J. Oh, and A. Stentz, “Grounding spatial relations for outdoor robot navigation,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1976–1982.

[125] F. Bernuy and J. Solar, “Semantic mapping of large-scale outdoor scenes for autonomous off-road driving,” in *IEEE International Conference on Computer Vision Workshops*, 2015, pp. 35–41.

[126] T. Vanderbilt, “Let the robot drive: The autonomous car of the future is here,” *Wired Magazine*, pp. 1–34, 2012.

[127] Technology in Our World: Whose (Augmented) Reality? [Online]. Available: http://infospace.ischool.syr.edu/2012/02/20/technology-in-our-world-whose-augmented-reality/

[128] The World Health Organization - Visual impairment and blindness. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/

[129] O. M. Mozos, C. Stachniss, and W. Burgard, “Supervised learning of places from range data using AdaBoost,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 1730–1735.

[130] A. Flint, C. Mei, D. Murray, and I. Reid, “A dynamic programming approach to reconstructing building interiors,” in *European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 394–407.

[131] Y. Tian, X. Yang, and A. Arditi, “Computer vision-based door detection for accessibility of unfamiliar environments to blind persons,” in *The 12th International Conference on Computers Helping People with Special Needs (ICCHP)*. Springer, 2010, pp. 263–270.

[132] F. Neves dos Santos, P. Costa *et al.*, “A visual place recognition procedure with a Markov chain based filter,” in *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2014, pp. 333–338.

[133] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel, “Towards semantic SLAM using a monocular camera,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1277–1284.

[134] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion: Real-time dense surface map-

ping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011, pp. 127–136.

[135] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke, "Dense real-time mapping of object-class semantics from RGB-D video," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 599–609, 2015.

[136] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from RGB-D images," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2631–2638.

[137] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3607–3613.

[138] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.

[139] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[140] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6555–6564.

[141] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," in *The 12th International Symposium on Experimental Robotics (ISER)*. Springer, 2010.

[142] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 719–722.

[143] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Robotics Research*. Springer, 2017, vol. 100, pp. 235–252.

[144] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1935–1942.

[145] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[146] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2174–2181.

[147] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6612–6619.

[148] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, pp. 1231–1237, 2013.

[149] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004, pp. I–I.

[150] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 3946–3952.

[151] P. Corke, D. Strelow, and S. Singh, "Omnidirectional visual odometry for a planetary rover," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, 2004, pp. 4007–4012.

[152] L. Kneip, M. Chli, and R. Y. Siegwart, "Robust real-time visual odometry with a single camera and an IMU," in *The British Machine Vision Conference*. British Machine Vision Association, 2011.

[153] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 486–492.

[154] T. D. Barfoot, "Online visual motion estimation using FastSLAM with SIFT features," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 579–585.

[155] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[156] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[157] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *The 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 225–234.

[158] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2320–2327.

[159] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1449–1456.

[160] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[161] P. Ren, W. Sun, C. Luo, and A. Hussain, “Clustering-oriented multiple convolutional neural networks for single image super-resolution,” *Cognitive Computation*, vol. 10, no. 1, pp. 165–178, 2018.

[162] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[163] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[164] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[165] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[166] K. Kawakami, “Supervised sequence labelling with recurrent neural networks,” Ph.D. dissertation, Ph. D. thesis, Technical University of Munich, 2008.

[167] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, “Spatial transformer networks,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[168] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.

[169] TensorFlow. [Online]. Available: https://www.tensorflow.org/

[170] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2366–2374.

[171] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[172] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 963–968.

[173] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[174] R. Li, Q. Liu, J. Gui, D. Gu, and H. Hu, "A novel RGB-D SLAM algorithm based on points and plane-patches," in *The 12th IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 1348–1353.

[175] I. Jebari, S. Bazeille, E. Battesti, H. Tekaya, M. Klein, A. Tapus, D. Filliat, C. Meyer, S.-H. Ieng, R. Benosman *et al.*, "Multi-sensor semantic mapping and exploration of indoor environments," in *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 2011, pp. 151–156.

[176] J. Wang, H. Zha, and R. Cipolla, "Combining interest points and edges for content-based image retrieval," in *The 12th IEEE International Conference on Image Processing (ICIP)*, vol. 3, 2005, p. 1256.

[177] T. Nicosevici and R. Garcia, "Automatic visual bag-of-words for online robot navigation and mapping," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 886–898, 2012.

[178] N. Kejriwal, S. Kumar, and T. Shibata, "High performance loop closure detection using bag of word pairs," *Robotics and Autonomous Systems*, vol. 77, pp. 55–65, 2016.

[179] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[180] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.