An Investigation into Image-based Indoor Localization using Deep Learning

Qing Li

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy

October 2019

Abstract

Localization is one of the fundamental technologies for many applications such as location-based service (LBS), robotics, virtual reality (VR), autonomous driving, and pedestrians navigation. Traditional methods based on wireless signals and inertial measurement unit (IMU) have inherent disadvantages which limit their applications. Although image-based localization methods seem to be promising supplements to previous methods, their applications in the indoor scenario have many challenges. Compared to the outdoor environments, indoors are more dynamic which adds difficulty to map construction. Also indoor scenes tend to be more similar to each other which makes it difficult to distinguish different places with similar appearance. Besides, how to utilize widely available 3D indoor structures to enhance the localization performance remains to be well explored.

Deep learning techniques have achieved significant progress in many computer vision tasks such as image classification, object detection, monocular depth prediction amongst others. However, their application to indoor image-based localization has not yet been well studied. In this thesis, we investigate image-based indoor localization through deep learning techniques. We study the problem from two perspectives: topological localization and metric localization. Topological localization tries to obtain a coarse location whilst metric localization aims to provide accurate pose, which includes both position and orientation. We also study indoor image localization with the assistance of 3D maps by taking advantage of the availability of many 3D maps of indoor scenes. We have made the following contributions:

Our first contribution is an indoor topological localization framework inspired by the human self-localization strategy. In this framework, we propose a novel topological map representation that is robust to environmental changes. Unlike previous topological maps, which are constructed by dividing the indoor scenes geometrically, and each region is represented by the aggregation of features derived from the whole region, our topological map is constructed based on the fixed indoor elements and each node is represented with their semantic attributes. Besides, an effective landmark detector is

devised to extract semantic information of the objects of interest from the smart-phone video. We also present a new localization algorithm to match the detected semantic landmark sequence against the proposed semantic topological map through their semantic and contextual information. Experiments are conducted on two test sites and results show that our landmark detector is capable of accurately detecting the landmarks and the localization algorithm can perform localization accurately.

The second contribution is that we advocate a direct learning-based method using convolutional neural networks (CNNs) to exploit the relative geometry constraints between images for image-based metric localization. We have developed a new convolutional neural network to predict the global poses and the relative pose of two images simultaneously. This multi-tasking learning strategy allows mutual regularizations for both the global pose regression and the relative pose regression. Furthermore, we designed a new loss function that embeds the relative pose information to distinguish the poses of similar images of different locations. We conduct extensive experiments to validate the effectiveness of the proposed method on two image localization benchmarks and achieve state-of-the-art performance compared to the other learning-based methods.

Our third contribution is a single image localization framework in a 3D map. To the best of our knowledge, it is the first approach to localize a single image in a 3D map. The framework includes four main steps: pose initialization, depth inference, local map extraction, and pose correction. The pose initialization step estimates the coarse pose with the learning-based pose regression approach. The depth inference step predicts the dense depth map from the single image. The local map extraction step extracts a local map from the global 3D map to increase the efficiency. Given the local map and generated point cloud, the Iterative Closest Point (ICP) algorithm is conducted to align the point cloud to the local map and then compute the pose correction of the coarse pose. As the key of the method is to accurately predict the depth from the images, a novel 3D map guided single image depth prediction approach is proposed. The proposed method utilized both the 3D map and the RGB image where we use the RGB image to estimate a dense depth map and employ the 3D map to guide the depth estimation. We show that our new method significantly outperforms current RGB image-based depth estimation methods for both indoor and outdoor datasets. We also show that utilizing the depth map predicted by the new method for single indoor image localization can improve both position and orientation localization accuracy over state-of-the-art methods.

Acknowledgments

Pursuing PhD degree is a touch process and I would like to thank many people along the way. Without them, I am not able to make it.

Foremost, I would like to thank my supervisor Prof. Guoping Qiu. I am sincerely grateful for your support and care not only on research but life as well. It is a privilege for me to take your guidance. Your open mind, patience and demand for perfection set an example for what a scientist should be. The tiles of paper drafts with your red comments on them will always remind me of to pursue the rigorous scholarship.

I would like to thank my second supervisor Prof. Jonathan .M . Garibaldi for the advices on my research and life convenience specially in UK. I am also grateful for my supervisors Prof. Qingquan Li and Jiasong Zhu for their insightful suggestions on research and life support in Shenzhen University.

I would like to thank all the Horizon CDT people. I have to thank Prof. Stedve Benford, Prof. Sarah Sharples and Dr. Michel Valstar for the interview and accepting me as a member of Horizon family. I would like to thank Emma Juggins, Dr. Sarah Martindale, Felicia Black, Andrea Haworth for your assistance for my research and life in the University. CDT cohort2015, it is a Honor to know you guys and I will value all the time we spent together.

I would like to thank VISTA lab people. Rui Cao, Bozhi Liu, Ke Sun, Xianxu Hou, Jingxin Liu, Bolei Zhou, Wei Pan, Jun Liu, Kanglin Liu, Wei Liu, Hui Yin, Wenming Tang, Hongming Luo, Guangsen Liao, Ruitao Xie, the insight discussion and happy time together will be life-long memory.

I would like to thank my friends. Yanjie Wang, Can Wang, Siyang Song, Bo Wang, Zixiao Shen, Jingjin Yan, Dan Wang, Tianlun Fei, Fei Yang, I will remember the great and hard time with you guys.

I would like to thank my family, my mother, father, younger sister and brother. I am grateful to be a family member and you are the best.

Publications

1. Qing Li, Jiasong Zhu, Tao Liu, Jonathan.M.Garibaldi, Qingquan Li, Guoping Qiu. Visual landmark sequence-based indoor localization[C]//Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery. ACM, 2017: 14-23.

2. Jiasong Zhu, Qing Li, Rui Cao, Ke Sun,Tao Liu, Jonathan.M.Garibaldi, Qingquan Li, Bozhi Liu, Guoping Qiu. Indoor topological localization using a visual landmark sequence[J]. Remote Sensing, 2019, 11(1): 73. [*Chapter*3]

3. Qing Li, Jiasong Zhu, Rui Cao, Ke Sun, Jonathan M. Garibaldi, Qingquan Li, Bozhi Liu, Guoping Qiu. Relative Geometry-Aware Siamese Neural Network for 6DOF Camera Relocalization. *Submitted to NeuroComputing* [*Chapter*4]

4. Qing Li, Jiasong Zhu, Jun Liu, Rui Cao, Hao Fu, Jonathan M.Garibaldi, Qingquan Li, Bozhi Liu, Guoping Qiu. Single Indoor Image Localization in a 3D Map. *Submitted to Journal of Photogrammetry and Remote Sensing* [*Chapter*5]

5. Rui Cao, Qian Zhang, Jiasong Zhu, Qing Li, Qingquan Li, Bozhi Liu, Guoping Qiu. Enhancing Remote Sensing Image Retrieval Using a Triplet Deep Metric Learning Network. International Journal of Remote Sensing.

6. Rui Cao, Jiasong Zhu, Qing Li, Qian Zhang, Qingquan Li, Bozhi Liu, Guoping Qiu. Learning Spatial-aware Cross-view Embeddings for Ground-to-Aerial Geolocalization. International Conference on Image and Graphics (ICIG) 2019.

Contents

List of Figures

List of Tables

Abbreviations

6DOF Six Degree of Freedom

BSB Business South Building

CDL Correctly Detected Landmarks

CN Corner

CNNs Convolutional Neural Networks

CSB Computer Scienece Building

DL Detected Landmark

DP Doorplate

DR Door

DTT Disabled Toilet Tag

ELV Elevator

EPA Environmental Protection Agency

FE Fire Extinguisher

GlobalLoss Comprehentive Loss

GPRU Global Pose Regression Unit

HMM2 Second Order Hidden Markov Model

ICP Iterative Closest Point

IMU Inertial Measurement Unit

LBS Location-based Service

MDLoss Adaptive Metric Distance Loss

MTT Men's Toilet Tag

PDR Pedestrian Dead Reckoning

RelLoss Relative Pose Loss

RelRLoss Relative Pose Regression Loss

RFID Radio Frequency Identification

RPRU Relative Pose Regression Unit

SfM Structure from Motion

SLAM Simultaneous Localization and Mapping

ST Stairs

UWB Ultra Wide-band

VLSIL Visual Landmark Sequence-based Indoor Localization

VR Virtual Reality

WDL Wrongly Detected Landmarks

WLL Wall

WMTT Women's toilet tag

CHAPTER 1

Introduction

According to the report from the Environmental Protection Agency (EPA), people spend above 90 percent of their time indoors. One of the basic needs of people in indoor environments is to know where they are. The purpose of indoor localization is to address such a problem. Various approaches and systems have been proposed based on wireless signals or inertial measurement unit (IMU). However, due to the strict requirements on deploying sensors for wireless signal-based methods and the decreasing accuracy of IMU-based methods over time, they are limited to certain scenarios.

Image-based indoor localization methods have attracted researchers' interests, and grow into a hot research topic as they do not have the limitations that previous methods face. Besides, they have been a vital component of many location-based applications, such as augmented reality, tourist navigation and movement tracking, as well as many computer vision tasks such as robotics, structure from motion (SfM) and simultaneous localization and mapping (SLAM). The image-based indoor localization problems can be categorized into topological and metric localization. Topological localization predicts a location that represents a region of scenes. A topological map is a graph and its nodes indicate the regions of the scenes. The topological localization intends to find the node where the images are taken. It is a highly abstract and sparse representation of the environment and is very suitable for navigation. It is also user-friendly and can be easily understood by the human being. Metric localization aims to estimate the exact position (X,Y,Z) as accurate as possible. The map of the scene is expressed by continuous coordinates. Metric localization is very important in many applications like robot navigation and virtual reality.

This thesis deals with the image-based topological and metric localization for indoor scenes. For topological localization, we intend to find an effective and understandable representation of nodes and study new localization algorithm for the new presentation.

For metric localization, we aim to develop fast and accurate image-based localization methods in two cases: without referenced 3D models and with referenced 3D models.

In the following sections, we first give a basic introduction to the wireless signal-based localization approaches, topological localization and metric localization as well as the convolutional neural networks in section 1.1. Then we illustrate the motivations behind this thesis and analyse the challenges of image-based indoor localization in section 1.2. Finally, we present our contributions to the solution to these challenges in section 1.3 and the arrangement of the thesis in section 1.4.

1.1 Background

1.1.1 Wireless Signal-based Indoor Localization

A number of technologies have been proposed for indoor localization by utilizing wireless signals such as WiFi [1–3], Blue-Tooth [4, 5], Ultra Wide-band (UWB) [6, 7], and radio-frequency identification (RFID) [8–10]. Three typical location estimation schemes are used to implement localization, including triangulation, scene analysis, and proximity. Triangulation estimates the target location through triangulating theory. Given two reference points in 2D space or three reference points in 3D space, the target location can be predicted by measuring its distance or angle direction to the reference points. The distance and the angle direction are estimated from the wireless signals strength based on the law of the signal propagation. These approaches have certain disadvantages. Distances or angle directions can not be accurately estimated from electromagnetic wave signal strengths as they are seriously influenced by the environmental change, leading to low accuracy. Besides, multiple path effect also brings the challenges to estimate the accurate distance and the angle direction .

Scene analysis is based on the retrieving strategy. The on-line measurement is matched to a number of pre-collected measurement with known locations, which are also called fingerprints. And the location of the querying measurement is attained by the location of the most similar fingerprint. Due to requirements of constructing numerous correspondences between locations and measurements, this type of methods is quite laborious and time-consuming. For large scenes, large storage needs to be satisfied to store the collected fingerprints. Besides, the matching process is quite inefficient as the querying measurement has to be compared with all the fingerprints. Furthermore, since the fingerprints and the querying measurements are captured with different devices at different times, an additional calibration is required to obtain good results.

Proximity algorithms provide a coarse location by identifying sensors with known positions. The location information is given with position of the detected unit or that of the strongest signal if multiple units are detected. These methods depend on the dense distribution of electronic sensors to obtain high accuracy, which could be costly. Besides, installing the electronic device in the indoor environment can damage the decoration. Pedestrian Dead Reckoning (PDR) approaches rely on an Inertial Measurement Unit (IMU) to measure the velocity and angle changes, and further estimate the position by accumulating them over time. Due to persistent influence of the measurement noise, the localization accuracy drops [11–13] over time.

1.1.2 Image-based Topological Localization

Topological localization is proposed in the late 1970s. It summarizes the real world as a graph, and the location is given by a node [14]. In this manner, the memory demand can be significantly reduced, hence it is very suitable for representing large-scale environments. The topological approaches allow robust performance against getting lost due to the multi-modal representation of locations.

Topological localization algorithms are developed based on the directed graph, which is also called topological map. In topological map, nodes indicate locations, and the arrow arcs represent the adjacency relationships from bottom location to the arrowed location. Each node is associated with a feature derived from the measurements of environments. The main task of topological localization is to correctly match the current captured information to the topological map based on certain similarity metric. Many devices have been utilized to construct the topological map, including the laser scanner, sonar, light, and wireless signal emitters et al. Among them, camera has been an important one, because it not only provides rich and robust visual appearance information but also is human-understandable.

In general, visual topological localization approaches can be divided into two categories: visual appearance-based approaches and landmark-based approaches. Visual appearance-based approaches take into considerations all the visual information and derive various visual features to present the location. Figure 1.1 shows the general procedure of visual appearance-based approaches. Landmark-based methods pay more attention to the salient and distinctive information and refer them as landmarks. They provide the locations information by identifying the landmarks. It mainly contains two steps: landmark detection and landmark recognition. Landmark detection searches the salient region of the images and the landmark recognition aims to identify the detected node in the topological map. Topological maps are constructed without metric

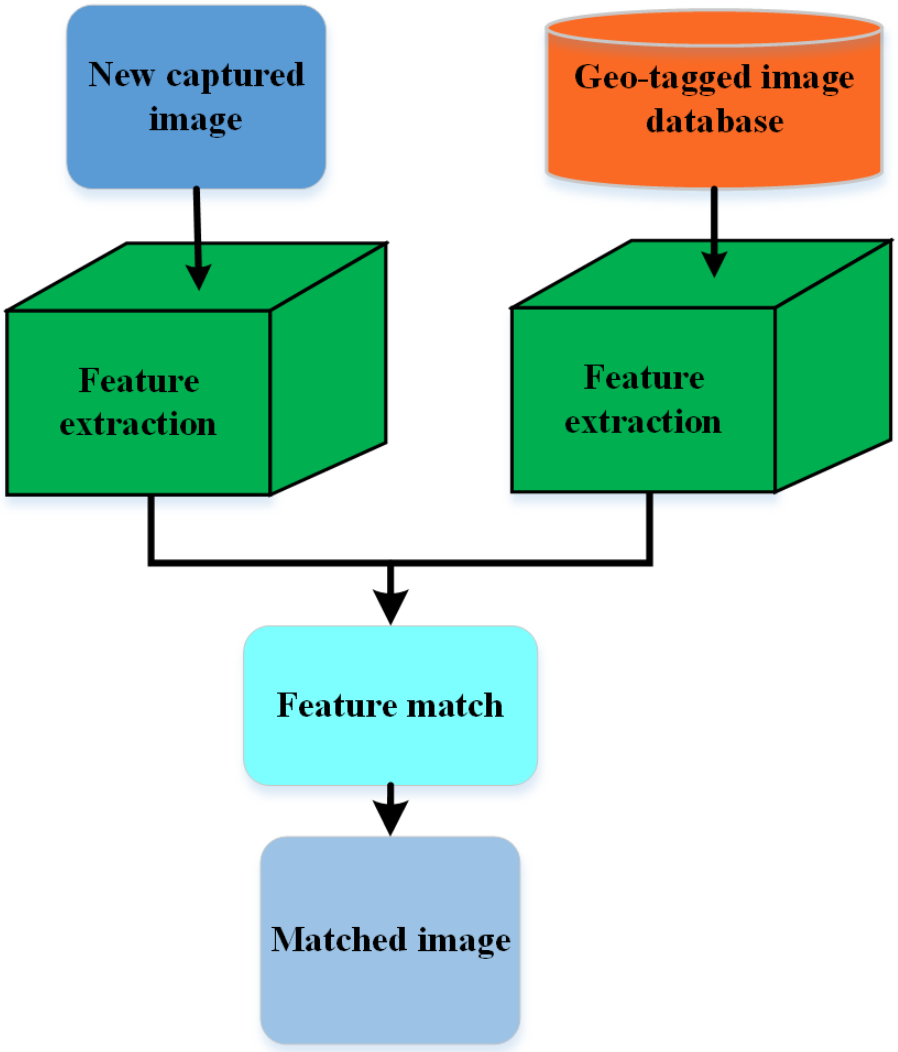


Figure 1.1: Demonstration of appearance-based localization framework. It includes two main steps: feature extraction and feature matching. The location of new captured image is estimated as that of matched location.

information, which means that the distance and directions between nodes can not be computed from it. However, metric maps contain such information.

1.1.3 Image-based Metric Localization

For metric localization methods, the environment is represented by a metric map in which every position is denoted by a coordinate with respect to certain coordinate system. On contrary to the topological localization, which provides a coarse localization result, metric localization obtains the exact position expressed with coordinates. Image-based metric localization tries to obtain precise location. Generally, it is achieved under three schemes: image retrieval-based methods, SfM-based methods, and learning-

based methods.

Image retrieval-based methods are very similar to the aforementioned appearance-based topological localization, which also is composed of feature extraction and feature matching. The main difference is that for appearance-based approaches, the visual feature is associated with a region of the scenes, while for image retrieval based metric localization, the feature is associated with a coordinate. In addition, the latter stores more features as metric map is a denser representation of the scenes.

SfM-based methods estimate the exact position by constructing the 2D-to-3D correspondences. 2D-to-3D associations follow the camera pinhole geometry as shown in Figure 1.2, which not only relates about the position and orientation of the camera, but also the camera internal parameters. Such geometric information can be expressed by equation (1.1.1).

$$\begin{bmatrix} x \\ y \end{bmatrix} = K \times \begin{bmatrix} R & T \end{bmatrix} \times \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix}, \tag{1.1.1}$$

where $[x, y]^t$ represents the coordinates of key point p on the image, $[X_W, Y_W, Z_W]^t$ indicates the coordinates of the corresponding 3D point P in the real word, K is the intrinsic matrix that is related to the camera, and R, T represent the orientation and position respectively. Generally, the internal parameters are known if the camera is given. Image-based metric localization assumes that each 3D points are associated with visual features, which are usually produced with SfM algorithms. Then the 2D image points can only be matched with 3D map points by comparing visual features. To solve the equation, at least three pairs of matched points are needed to estimate the position and orientation of the captured image. Considering that there exist the wrong matching pairs, the RANSAC algorithm [15] is utilized to filter them to achieve high accuracy. However, this scheme is vulnerable to inaccurate 2D-3D matches and suffers from expensive computation, especially in large-scale environments.

Learning-based methods utilize machine learning techniques to model the latent relationship between images and positions. It takes an image as input and directly estimates the position.

1.1.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) are one of the special types of neural networks, which have been widely used for many computer vision tasks such as image classifications, objects detection, and face recognition. A typical CNN consists of a series of

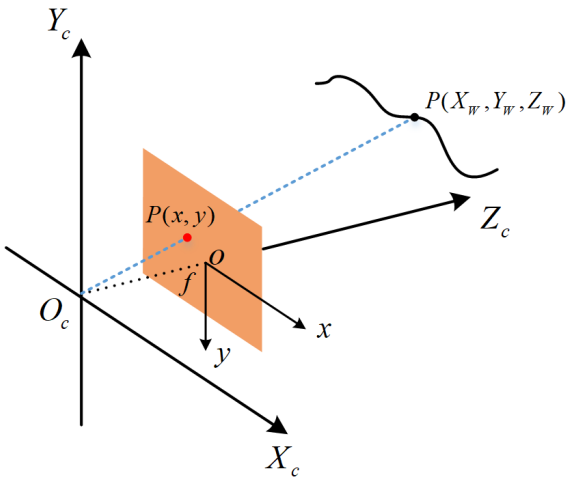


Figure 1.2: The projection geometry between the 2D image and 3D scenes. The (X_c, Y_c, Z_c, O_c) represents the coordinate system of the real world scene. The xoy indicates the image 2D coordinate system. $P(x, y)$ and $P(X_W, Y_W, Z_W)$ are corresponding points from 2D images and 3D scenes.

convolutional operation, pooling operation, fully connected layer, and a classifier to estimate the probability (between 0 and 1) of an object. Figure 1.3 gives an example of the CNN.

Regular neural networks also are referred to the multilayer perceptions, where each neuron in one layer is connected to all neurons in the next layer. Regular neural networks believe that each neuron is related to the whole feature map of previous layers while CNNs simplify it with local connection and assemble the information with more convolutional layers. It is also helpful to avoid over-fitting. Since the feature is more related to its neighbours, CNNs achieve better results than the regular ones on many computer vision tasks. The filter is much smaller than regular neural networks due to the local connection, thus it is more efficient.

Another property of CNNs is parameter sharing. In regular network, the weights of filters are tied with the input while CNN filters slide over the whole input image with the same parameters. The last attribute is the pooling strategy. CNNs contain many pooling operations, which extract the statistical information from the local regions. For instance, *max-pooling* function finds the maximum value to represent the local area. Besides, pooling operation makes it invariant to the slight transition of the input since it derives dominant information from a region.

The convolutional module is the central component of CNNs, which consists of three main operations: convolution, activation, and pooling operation. In the context of a convolutional neural network, convolution operation is a linear operation that involves

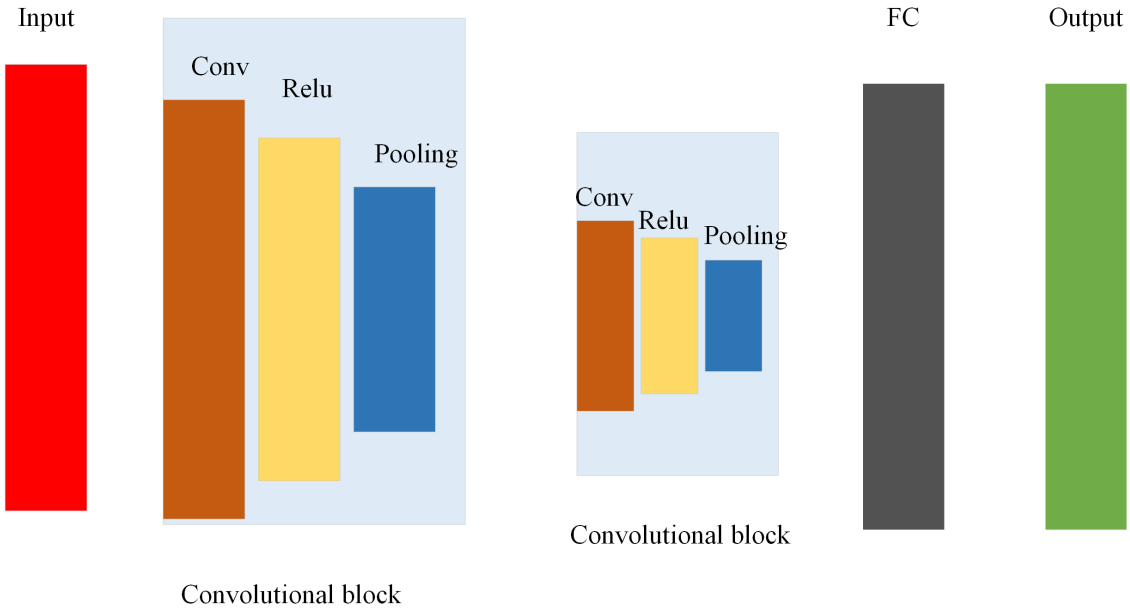


Figure 1.3: Example of the CNN architecture. It is composed of an input layer, an output layer, a fully connected layer (FC) and two consecutive convolutional blocks, which are consisted of convolutional layer (Conv), relu layer (Relu) and pooling layer (Pooling).

the multiplication of a set of weights with the input. Given that the technique is designed for two-dimensional input, the multiplication is performed between an array of input data and a two-dimensional array of weights, called a filter or a kernel. The filter is smaller than the input data and the type of multiplication is applied on a filter-sized patch of the input and the filter, thus it is a dot product.

The convolutional operation has four parameters: kernel size, stride, padding and dilation. Kernel size determines the receptive field of the filter. The larger it is, the more neurons are used for convolution. The stride indicates the step size when the filter slides over the input, which usually is one to compute for the every position of the input. When the filter reaches the boundaries of the input, the part of it will be out of the data. Padding parameter indicates whether the boundary region is computed or not. If it is true, the input will be extended with zeros, otherwise, the filter will not slide over the boundary position. To enlarge the receptive field without increasing the computational complexity, dilated convolution is used. Instead of computing all the pixel in the local region, dilated convolution computes it at intervals of T position. T is the parameter related to the receptive field, and the large it is, the larger receptive area is.

In a neural network, the activation function is responsible for transforming the sum of weighted input from the node into the activation of the node or output for that input.

Sigmoid and *tanh* functions are used to serve as activation functions for regular neural networks. They are not suitable for CNN because they saturate, which means that large values snap to 1.0 and small values snap to -1 or 0 for *tanh* and sigmoid respectively. Besides, they are sensitive to changes around their mid-point of their input, such as 0.5 for sigmoid and 0.0 for *tanh*. The limited sensitivity and saturation of the function happen regardless of whether the summed activation from the node provided as input contains useful information or not. Once saturated, it becomes challenging for the learning algorithm to adapt the weights to improve the performance of the model. Finally, very deep neural networks using sigmoid and *tanh* activation functions are difficult to be trained.

In order to back-propagate errors to train deep neural networks, an activation function is needed that looks and acts like a linear function, but is a non-linear function allowing complex relationships in the data to be learned. The function must also provide more sensitivity to the activation sum input and avoid easy saturation. The rectified linear activation function is a simple calculation that returns the value provided as input directly, or the value 0.0 if the input is 0.0 or less. Because rectified linear units are nearly linear, they preserve many of the properties that make linear models easy to optimize with gradient-based methods.

The deep layers can not guarantee the better performance as in large networks more layers using these non-linear activation functions fail to receive useful gradient information. Error is back-propagated through the network and used to update the weights. The amount of error decreases dramatically with each additional layer through which it is propagated, given the derivative of the chosen activation function. This is called the vanishing gradient problem and prevents deep (multi-layered) networks from learning effectively.

Pooling layers can reduce the number of parameters when the images are too large. Spatial pooling also called sub-sampling or down-sampling, which reduces the dimensionality of each map but retains the important information. Common spatial pooling includes Max Pooling, Average Pooling, and Sum Pooling. Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map is called as sum pooling. The fully connected layer is much like the regular neural network, in which we flattened the feature map of the convolutional layers into a vector and feed it into a fully connected layer to further aggregate the information.

1.2 Motivations and Challenges

Indoor localization is one of the fundamental components for location-based service and many other applications such as navigation, robotics and virtual reality. It aims to estimate the position given a map of the indoor scene. The position can be represented by a node of the topological graph, or a coordinate of a metric map. Unlike the outdoor environment, the well-developed GPS techniques endow us the high real-time position in the outdoor environment. However, GPS-based technologies fail to work in the indoor scenes as GPS signals are blocked or weakened to perform localization. Many researchers proposed to exploit wireless signals like WiFi, mobile phone signals, and ultrasonic band sound to address the problem based on triangulation or fingerprints matching. The main drawback of them is their vulnerability to the environment change due to moving pedestrians or furnitures, which occurs frequently in the indoor environments. Furthermore, these methods are expensive and labour-extensive for large-scale deployments and suffer from discontinuous tracking during pedestrian movement.

Due to the limitations of previous methods, image-based localization grows into a popular solution to indoor localization problems. It is attractive mainly for two reasons. First, cameras have been an essential component of mobile phone, thus no special equipment is needed to collect information compared with RFID-based methods. Another important reason is that image-based localization approaches utilize the natural appearance of the indoor scenes and have no requirement to change the infrastructure of the indoor scenes. Furthermore, the availability of 3D indoor models can be utilized to further boost the image-based localization performance.

Many image-based localizations are proposed and have shown their effectiveness in the outdoor environment. They mainly depend on structure from motion techniques or image retrieval techniques. When it comes to the indoor case, they are likely to fail. It is because the surfaces of indoor scenes are usually textureless and repetitive, and indoor scenes are of high similarity. It is not feasible to extract enough key points from textureless surface to perform structure from motion and similar appearance of the indoor scenes confuses image retrieval-based methods to correctly localize the query image and lead to mismatches of SfM-based methods.

For indoor topological localization, two main problems are the place similarity and dynamic environment change. Unlike outdoor environments, indoor scenes are of similar structure and decoration, which results in the high visual similarity of different locations. The locations can not be distinguished with visual appearance only. Another

problem is that the indoor environment changes more frequently, which causes the appearance changes at different time. How to derive the robust representation of the locations remains a difficult problem.

Conventional image-based localization methods perform metric localization by constructing the 2D-3D correspondences. It works well in the outdoor environment, as outdoor scenes contain complex texture. On the contrary, indoor scenes usually are textureless and repetitive, making it difficult to detect key points and extract local feature. Besides, the blur caused by camera motion and image occlusion also influence the accuracy of such methods. Another challenge is that it has to store a number of 3D points, which is storage overhead and inefficient to perform match.

Deep learning-based methods have shown the potential to solve the aforementioned problems. A deep convolutional neural network is trained with a loss formulated from the difference of predicted poses and targeted poses. However, this type of methods fail to distinguish the image of high similarity. It is mainly caused for two reasons: firstly, the indoor images are of high similarity; secondly, the down-sampling effect of convolution operation neglects the tiny differences between images.

Traditional image-based metric localization methods construct the 2D-3D match through local features based on the 3D models built through structure from motion techniques. It is difficult to build indoor 3D models with structure from motion techniques which is still due to the textureless surface of indoor scene. With the development of LiDAR sensors such as Kinect and various types of LiDAR scanners, many 3D models have been built for indoor scene with them instead of using SfM algorithm. However, it still remains a challenge to localize images in 3D LiDAR map. Unlike 3D model built from structure from motion (SfM) where each 3D point is associated with a local image feature, 3D LiDAR model only contains geometric information. The key challenge is to bridge the gap to match 2D geometry and 3D geometry.

1.3 Contributions

In this thesis, we make the following contributions:

- Visual Landmark Sequence-based Indoor Localization (VLSIL) framework for topological localization. We propose a novel Visual Landmark Sequence-based Indoor Localization (VLSIL) framework to acquire indoor location through smartphone videos. We propose a novel topological node representation using semantic information of indoor objects. Moreover, we present a robust landmark de-

tector using convolutional neural network for landmark detection that does not need to retrain for new environment. We present a novel landmark localization system built on a second order hidden Markov model to combine landmark semantic and connectivity information for localization, which is shown to relieve the scene ambiguity problem where traditional methods have failed.

- Relative geometry aware Siamese neural network for image-based metric localization. We present a novel relative geometry-aware Siamese neural network to enhance the performance of deep learning-based methods through explicitly exploiting the relative geometry constraints between images. We perform multi-task learning and predict the absolute and relative poses simultaneously. We regularize the shared-weight twin networks in both the pose and feature domains to ensure that the estimated poses are globally as well as locally correct. We employ metric learning and design a novel adaptive metric distance loss to learn a feature that is capable of distinguishing poses of visually similar images from different locations. We evaluate the proposed method on public indoor and outdoor benchmarks and the experimental results demonstrate that our method can significantly improve localization performance. Furthermore, extensive ablation evaluations are conducted to demonstrate the effectiveness of the different terms of the loss function.

- Single image localization in 3D map framework. Unlike previous methods that require the 3D map to be constructed from a number of RGB images using structure from motion , our 3D map is composed of 3D point cloud without any colour features. We propose a new framework to address the problem by performing geometry matching in the 3D space. The framework consists of four main stages: initial pose estimation, local map extraction, depth prediction, and pose refine through Iterative Closest Point algorithm (ICP) [16]. We exploit the deep learning-based single image depth prediction to generate the dense depth map, and simply transform it into the 3D point cloud with the camera intrinsic parameters and initial pose prediction. Given the generated 3D point clouds, we register the generated 3D point clouds into the 3D map. To reduce the time cost on searching corresponding points, we use local 3D map extracted from the global 3D map as alternative to the global map to perform 3D match. The key component of proposed method is to accurately predict the dense depth map. To address it, we propose a novel 3D map guided single image depth prediction method. The method is also based on the convolutional neural network. Instead of taking a single RGB image as input , we feed RGB image initial depth into the network to

handle the scale ambiguity problem that RGB-based depth prediction methods always suffers from. Initial depth is generated by projecting the 3D point cloud into the coarse localization results from other localization algorithms. We conduct experiments on both indoor and outdoor benchmarks and the results show that our method outperforms RGB-based methods over RGB image depth prediction. We also conducted experiments to show that the proposed method can increase the localization performance than the learning-based methods.

1.4 Outline

This thesis is composed of six chapters. Chapter 1 provides the background information on wireless indoor localization, image-based topological localization, metric localization and the deep convolutional neural networks. The challenges and motivations are illustrated as well. We also elaborate the main contributions of the thesis in this chapter. Chapter 2 reviews the image-based localization methods and related techniques on both topological localization and metric localization along with the related techniques for single image localization in 3D maps. Chapter 3 describes our new topological localization framework with semantic landmark sequence for indoor environments in detail. We also illustrate the proposed topological map and localization algorithms in this chapter. Chapter 4 elaborates the proposed novel convolutional neural network for direct pose regression by exploiting the relative geometry constraints. Chapter 5 introduces our single image localization in 3D map framework as well as RGB image depth prediction approach. Finally, we conclude the thesis in chapter 6.

CHAPTER 2

Literatures and Methods

This thesis aims to address the image-based indoor localization problem with regard to the topological localization, and metric localization with and without 3D models assistance respectively. Therefore, in this chapter, we give an overview of related methods and literatures in two tasks to clarify the research gaps. Besides, we also review the related works on monocular image depth prediction and 3D matching, which are vital components for single image indoor metric localization in a 3D map.

This chapter is organized as follows: Section 2.1 reviews current methods for image-based topological localization over the place representation and matching strategies, and discusses the limitations. Section 2.2 reviews image-based metric localization in three categories: image retrieval-based methods in section 2.2.1, SfM-based methods in section 2.2.2, and learning-based methods in section 2.2.3. Their advantages and disadvantages are analysed respectively. We also review the methods on monocular image depth prediction in section 2.3 and 3D matching in section 2.4, which are essential components for single image localization in 3D space.

2.1 Image-based Topological Localization

2.1.1 Place Representation

The place representation, which is also called node description, aims to characterize the place with discriminant information according to visual appearance. It should be representative among nodes, and fast to compute and compare. It can be categorized into two groups: appearance-based representation and landmark-based representation.

Appearance-based representation takes all the visual information of the location into consideration and represents it with various visual feature descriptors. One of widely

used descriptors is the local feature. Local features are computed at pixel level within a local neighbourhood of key points in the image. It mainly contains two steps: key point detection and feature description. Key point detection finds the salient points in the image, and the feature description depicts the key points based on their neighbourhood region. SIFT features have been widely used for place representation. It utilizes the Hessian-affine detector [17] to detect the salient points and is described with SIFT [18] descriptor. Many important descriptors have been devised based on it. RootSIFT [19] is introduced and achieves better results in matching step by slightly adding computational load. SURF descriptor [20] relieves the computational overhead and attains real-time performance in [21–23]. To decrease the storage demand, Feng et al. [24] employ binary BRISK descriptor [25] at the expense of the slight precision decrease.

Aforementioned local features describe the place according to the colour and gradient information, while primitive geometric shapes like line and contour can also be exploited to describe the place. For example, vertical lines have been widely used for building representation in urban environments [26–28]. Contour is employed to obtain the pose of the captured image in [29]. Normal vectors or planar surfaces have been extracted from point clouds generated from image to represent the places in [30, 31]. Besides, they can also be combined with other descriptors to present the place.

Another type of feature is global feature. It takes consideration of information from the whole images. Compared to the local features that require numerous descriptors to depict the place, the global features generate one high dimensional feature vector to represent the locations. Although it is less robust to the view perspective differences, it is still favoured as it is fast to compare and decrease storage requirement.

GIST descriptor [32] is one of the most used global descriptor for place representation [29, 33, 34]. It convolves the images with Gabor filters at different scales and orientations, and aggregates all the information to a vector. The raw image can also serve as a global descriptor, and perform the pixel-wise comparison [34, 35]. Histogram derived from colour [36], or depth [37] can also be viewed as a global feature. There are also researchers who transform the images into frequency domain through Fourier Transform (FT) to represent the place [38].

With the recent emergence of deep CNN, they have been introduced to learn global representations for images. Their high capacity has boosted the the performance of urban image retrieval [39–42]. The simple way to create the global feature is to use the output of the fully-connected layers. The global feature can also be generated by combing the feature maps from different layers with weights into a vector [43]. Many researchers have shown that features extracted from mid-level convolutional layers

achieve better performance than that of fully-connected layers [43, 44].

Appearance-based representation pays attention to all visual information of the locations, and is capable of discriminating different locations. However, the matching performance drops significantly due to the slight environmental changes, which often happen in indoor scenes. On the contrary, landmark-based representation utilizes the discriminant information to represent localization. This method is suitable for indoor scenes as background information has no impact even bad impact on place representation.

Visual landmarks can be divided into two categories: artificial landmarks and natural landmarks. Artificial landmarks are purposefully designed to be salient in the environment. Ahn et al. [45] design a circular coded landmark that is robust with perspective variations. Basiri et al. [46] develop a landmark-based navigation system using QR codes as landmarks and user's location is determined and navigated by recognizing quick response code registered in the landmark's location. Briggs et al. [47] utilize self-similar landmarks based on barcode and is able to perform localization in real-time. Artificial landmarks can be precisely detected since they are manufactured based on prior rules. Those rules allow them to stay robust facing challenges of varying illuminations, view points and scales in images and help to devise the landmark detectors. Their position can also be coded in the landmark appearance. However, deploying artificial landmarks changes building decoration which might not be feasible due to economic or owners' tastes. Natural landmarks avoid changing indoor surface by exploiting physical objects or scenes in the environment. Common objects like doors, elevators and fire extinguishers are good natural landmarks. They remain unchanged in a relatively long period and are common in the indoor environment.

Many methods have been proposed to represent locations using natural landmarks [48–50]. Some of them are based on hand-crafted features, which are devised to make use of colour, gradient or geometric information. Planar and quadrangular objects are viewed as landmarks and are detected based on geometric rules [48, 49]. Tian et al. [50] identify indoor objects like doors, elevators and cabinets by judging whether detected lines and corners satisfy indoor object shape constraints. SIFT feature is chosen to perform natural landmark recognition in [51, 52]. SerrÃčo et al. [53] propose a natural landmark detection approach by leveraging SURF feature and line segments. It performs well in detecting doors, stairs and tags in the environment. Kawaji et al. [54] use omnidirectional panoramic images taken from different positions as landmarks and PCA-SIFT is applied to perform image matching. Besides, shape [55, 56], light strength [57] or region connection relations [58] are also exploited to represent as landmarks for

localizations. Kosmopoulos et al. [59] develop a landmark detection approach based on edges and corners.

2.1.2 Matching Methods

When the amount of data is acceptable, brute-force match approach can be employed to find the exact nearest feature. This is usually used for the case when a single vector is used to describe a place.

The most common matching method is to compute the distance between two feature descriptors. For example, [39, 40] represent the location with the global feature trained by CNN and match the query descriptor against every visual feature in the database. Brute-force comparison is performed in [60, 61], where the place is represented with local features or hybrid features. Graph matching is also exploited for feature matching [62]. In this case, the query image is presented with a graph, in which the nodes are described with the visual words that are defined in advance and the edges denote the co-visibility of two words in an image. This formulation allows the integration of geometric relationship between visual words. A graph kernel is chosen to compute the similarity between the query image and the database. Note that graph-based approaches are often employed when scenes are described by spatially organized semantic clues such as office furnitures [63] and street equipments [64]. Area correlation algorithms can also be used for computing feature similarity. The sum of two corresponding patches are used to indicate the similarity of two image features [65, 66]. Wan et al. [38] propose a Phase Correlation on images that are transformed after Fourier transformation in order to relieve shadow artefacts.

Brute-force comparing scheme becomes unacceptable when the amount of the features is too large, especially for local features where each place consists of numerous local features. Many approaches try to find the approximate nearest neighbour to trade precision for efficiency at the expense of certain errors of retrieved results. Hashing methods [67] and quantization frameworks [68–70] are two common strategies to be used.

Machine learning techniques have been utilized to relieve the computational overhead by grouping the database images. SVM classifier has been widely used in many works [71–74]. In [72], database images are initially clustered into several groups based on the visual similarity of the images. For each group, a SVM classifier is trained for each cluster to determine which cluster the query image belongs to. During the querying time, the query image is fed into all the classifiers to find the best matched group. Then, the

query image will be compared with the images in the matched group. [72, 73] train linear classifiers over HOG descriptors to robustly find the similar images. Aubry et al. [71] utilize linear discriminant analysis (LDA) data representation instead of SVM for efficiency reason. Kim et al. [75] use SVM classifier to predict the confidence of extracted descriptors, which improves the matching efficiency by reducing the number of features comparing against the database. [37, 76] localize an input query among a set of predefined places by embedding the recognition process into probabilistic framework.

Another strategy to relive computational requirement to perform feature dimensional reduction. Dimension reduction of descriptor is often performed to reduce matching time and memory footprint. The most used technique is the principal component analysis (PCA). In [39, 40], PCA is applied on high dimension vector extracted from CNN layers. PCA has also been used to reduce the size of local features aggregated vectors [75] and global descriptors [37]. Gaussian Random Projection is applied in [77, 78] and in a different work, binary locality-sensitive hashing [44] is used instead [79]. To reduce data redundancy, various pooling strategies could be applied to final features before the similarity search [39, 40, 42, 44, 80, 81].

Many positioning algorithms have introduced landmarks for indoor localization. Basically, landmarks are taken as supporting information to reduce the error drift of dead reckon approaches [82–84]. It can also be utilized for indoor topological localization since landmarks play an important role in localizing and navigating pedestrians in an unfamiliar environment [85]. Many approaches perform landmark-based localization under geometric scheme. Triangle intersection theory is applied to localize users using more than 3 landmarks [86]. Another type of landmark-based localization utilizes the landmark recognition techniques. It assumes that users are near to the detected landmarks. The landmark is identified based on their visual representations [48, 49, 57]. However, in indoor environment, it is usually not feasible to match landmarks just based on visual feature, due to numbers of locations of the similar appearance. Additional information is needed to distinguish different landmarks. Tian et al. [50] exploit text information around doors to address the problem. However, it is not always possible to have tags of text around doors. Contextual information between landmarks is exploited through hidden Markov model (HMM) to recognize landmarks and achieves good results in [87–89]. However, HMM model only takes one previous landmark to recognize current landmark, and it fails in scenes of high ambiguity.

2.2 Image-based Metric Localization

Many image-based metric localization have been proposed to use smart-phone camera for indoor localization [54, 90–102]. These methods exploit computer vision techniques to estimate people's location and mainly fall into three categories: image retrieval-based methods, SfM-based methods, and deep learning-based methods.

2.2.1 Image Retrieval-based Methods

Many approaches and systems are proposed based on image retrieval technique [21, 103–112]. They determine the pose of the query image by matching it with images rendered from 3D scene models. The key component of the technique is image representation. Global descriptors are often used, such as colour histogram [113] and gradient orientation histogram [114]. GIST descriptor [32] and GIST-based descriptors [115] are applied to represent panoramic images in [116–118]. SeqSLAM [119] generates the global descriptors from a sequence of consecutive images instead of a single image. Global descriptors are fast to compute, but they are not robust to occlusion and illumination changes. Local features like SIFT [120] and SURF [20], have been used in [104] for image representation. Compared with the global descriptors, they are less sensitive to occlusion and view variations. However, the storage requirement of the method is high for large scenes. The pooling features like BoW [121] and VLAD [81] are able to relieve the challenge. They aggregate local features and represent the locations with a compact feature vector instead of a large number of local features [103]. Image retrieval-based methods use images captured by the smart-phone camera to search for similar images in the image dataset whose positions and orientations are already known. The pose of the query image is determined with poses of the similar images. This approach not only requires significant offline processing but can also easily get stuck in the situations where different locations have similar appearance.

2.2.2 SfM-based Methods

Another type of methods solves the problem by utilizing camera projection geometry between 2D pixels and 3D models. They estimate the pose by constructing the correspondence between 2D pixels and 3D points of the scene [122–126]. Local point features, like SIFT [120], SURF [20] and ORB [127], are frequently used to describe the detected 2D points. 3D points, generated using the SfM technique, are also described with local features to perform 2D-3D matching. It can achieve accurate results when

enough correct pairs are provided. The main challenge is to establish enough correct 2D-3D correspondences, which is difficult for two reasons. Firstly, local feature descriptor fails when a scene has repetitive texture or texture-less surface; and secondly, the process is inefficient for large scenes.

To increase the efficiency of the 2D-3D matching, prioritized search approaches [124, 125] are proposed to construct enough matching pairs instead of matching all detected 2D points. Scene coordinate random forest (SCRF) [128, 129] utilizes machine learning techniques to directly predict 3D coordinates of image pixels by training a random forest. Similar to SCRF, deep learning technique is employed to predict 3D coordinate of the centre point of an image patch in [130]. However, these methods require 3D model for the network training, which limits their application. To filter out the wrong matches, co-visibility information is exploited in [123, 124]. However, they do not work in low texture environments and they also suffer from image blurring caused by camera motion. In addition, environment change significantly decreases the performance of the two types of methods, which frequently occurs in the indoor environment.

2.2.3 Deep Learing-based Methods

Deep learning has achieved extraordinary performance in image classification, object detection, and image retrieval tasks. Many researchers have employed it to solve the camera relocalization problem [131–139]. PlaNet [131] regards the problem as a classification task. It divides the map into grids and predicts the grid in which the query image belongs to through deep learning technique. Many other researchers consider it as a regression problem instead. They directly estimate the pose through a convolutional neural network. PoseNet [132], built on the GoogLeNet model [133], is the first attempt to adopt this paradigm in an end-to-end manner. It is further extended to Bayesian PoseNet [134] to estimate the confidence of the result as well. HourglassNet [135] utilizes the encoder-decoder network structure with skipped connections to aggregate features from both lower and higher layers for pose regression. It achieves better performance than PoseNet. LSTM-Net [136] believes that high dimensional output of fully connected layer in PoseNet is not optimal. It adds a LSTM- network after the last fully connected layer in PoseNet to reduce information redundancy. VidLoc [137] exploits smooth constraints of a video to address the perceptual aliasing problem. It takes a video clip as input instead of a single image and proposes a bidirectional recurrent neural network structure to fuse the previous and next images information to increase predicted pose accuracy. Laskar [140] proposes a new triangulating strategy that predicts the pose by estimating the relative pose between the query image and the

images in the database. Its main drawback is low efficiency since the relative pose of all the images in the database has to be computed. PoseNet2 [138] introduces the re-projection error with global pose error and improves the performance. However, 3D points are required in their method. MapNet [139] fuses the inertial information with image information through deep learning to enhance the network performance.

The proposed method in this thesis is also based on convolutional neural networks. However, it has a number of distinctive features. For example, we use an innovative Siamese network architecture to exploit the relative geometry of images in addition to predicting the absolute poses. Unlike [138] and [139], we only rely on the 2D images for training. Compared to [134, 134–136], we take a pair of images as input and utilize their relative pose error for training. In contrast to [140], we directly regress the image pose instead of performing triangulation.

A very recent work that also uses multi-task learning and explicitly models relative poses of two frames appears in [141, 142]. However, our system architecture differs from that of [141, 142] in a number of significant ways. Whilst we use a Siamese network and metric learning loss to model the relative geometrics of two frames, [141, 142] use two separate networks to model the relative geometrics of two consecutive frames (Although [141, 142] refer their two networks as Siamese network, strictly speaking it is not a Siamese network architecture because the two networks do not share weights). Furthermore, while our method can model the relative geometrics of two arbitrary frames, but [141, 142] can only model two consecutive frames.

2.3 Monocular Camera Depth Prediction

Predicting depth of a scene can be obtained from multi-view stereo (MVS), structure from motion (SfM), optical flow, shape-from-shaping (SfS), simultaneous localization and mapping (SLAM), and shape from defocus [143]. However, these methods need more than one image and large resources. Therefore, researchers have recently pay more attention to a challenging per-pixel prediction task, i.e. monocular depth estimation, where there is only a single still RGB input image. Up to now, a variety of methods have been proposed, which can be divided into two categories: hand-crafted features based methods and deep neural network based methods.

2.3.1 Hand-crafted Features-based Methods

Early works on the monocular depth estimation have mainly used hand-crafted features (HOG [144]). Saxena et al. [145] firstly leverage learning approach to tackle this problem based on hand-crafted features, and employ a discriminatingly trained MRF to incorporate multi-scale local and global features. Saxena et al. [146] extend the previous model to infer 3D scene structure from a single image by over-segmenting the input image to many homogeneous regions using superpixels. Hoiem et al. [147] group superpixels to multiple constellations for constructing a 3D model of the single input image. Following this work, Liu et al. [148] exploit a simple MRF model to perform the monocular depth estimation from predicted semantic segmentation labels. Being aware that depth estimation and scene semantic labelling are closely tied to the property of perspective geometry, Ladicky et al. [149] propose a pixel-wise classifier to jointly predict the depth labels and semantic labels from a single image. Except for the above parametric approaches, non-parametric approaches are also used to estimate monocular depth. Karsh et al. [150] exploit RGBD datasets to match the input image by using SIFT Flow, and then make a global optimization procedure to produce a monocular depth map. Likewise, Liu et al. [151] utilize the availability of RGBD images to perform monocular depth estimation, while they formulated this task as a discrete-continuous optimization problem.

However, these traditional approaches heavily rely on hand-crafted features, and need preprocessing or post-processing operation. They are not suitable for complicated scenes, and are shown to be inefficient and ineffective.

2.3.2 Deep Neural Network-based Methods

Thanks to the emergence of publicly available RGBD datasets, DCNN has been shown to be a more efficient and high-qualified method for monocular depth estimation. Eigen et al. [152] firstly utilize DCNN for monocular depth estimation, which incorporates a coarse-scale network (based on AlexNet [153]) and a fine-scale network. A skip connection between two networks is used to refine the original prediction of the coarse-scale network. Moreover, they firstly exploit a scale-invariant loss function to regress this depth problem. Then, they extend this model to three disparate tasks (depth, surface normals, and semantic label) based on a deeper CNN (VGG-Net [154]), and three-scale skip connections are used to further refine the output [155]. Laina et al. [143] use a much more deeper architecture (ResNet-50 [156]) to infer depth, and further use up-projection blocks to attain a high-resolution output depth map. And, they firstly adopt

Berhu loss to reduce the long-tail effect of depth values. Deviating from these depth regression approaches, some works [157, 158] have treated monocular depth estimation as a classification task, and also achieved high performance. However, these methods cause the mosaic artefacts due to the discretization of the continuous depth.

The aforementioned methods have demonstrated that DCNN is good at extracting the deep global features. However, it is weak in attaining the fine-grained depth map due to a series of pooling layers and stride operations. Recently, some works [159–161] have demonstrated that skip-connections between the encoder and decoder can produce fine detail predictions in the pixel-level tasks. Nevertheless, there is a gap between the RGB image and the depth map, as mentioned in the previous section. Therefore, the straightforward skip connection between the RGB features in the encoder and the depth features in the decoder brings much noise in the output depth map. Besides, CRF is believed to be another effective way to enhance the details of the depth map. Since Lafferty [162] exploit CRF for segmenting and labelling sequence data, the CRF has attracted a lot of attention in the past two decades. In the early stage, CRF is regarded as a post-processing operation. Due to the efforts of Krahenbuhl et al. [163], Zheng et al. [164] and Teichmann et al. [165], they embed the CRF as a module in CNN, whose parameters can be learned by the training of the deep neural network. In order to improve the details of the depth map, Li et al. [166] introduce a hierarchical continuous CRF as post-processing to refine the depth output from CNN. Liu et al. [144] design a continuous CRF loss layer with super-pixels for CNN to better refine the depth prediction. Xu et al. [167] propose a CNN implementation continuous CRF to aggregate multi-scale features from the decoder of CNN. However, CRF mainly uses the original colour and spatial information of the RGB image to constrain the depth map, which will bring much irrelevant information to the depth map.

All the aforementioned methods belong to supervised learning methods that require the ground truth labelling of the depth. Some works exploit the unsupervised learning to predict the depth to avoid the demand of ground truth depth label. Garg et al. [168] use stereo pairs to train a network to predict the depth with the loss function formulated from the photometric difference between the true right image and synthesized one generated from the left image and the predicted depth. Godard et al. [169] improve the depth estimation by introducing the symmetric left-right consistency loss. Kuznietsov et al. [170] propose a semi-supervised learning framework by using sparse depth maps for supervised learning and dense photometric error for unsupervised learning. Zhou et al. [171] propose an approach which jointly predicts the image depth and its pose in a single network.

Additional information is also exploited with RGB data to perform depth prediction using a convolutional neural network. Ma et al. [172] predict full resolution depth from a few depth samples and images. Liao et al. [173] utilize sparse laser scanner points to aid RGB image depth prediction. Cadena et al. [174] propose a network to learn depth from the RGB image as well as the semantic labels. Zhang et al. [175] generate dense depth map by taking RGB-D image as input.

2.4 3D Localization

Traditional methods predict the location of the query image in a 3D map through establishing 2D-3D correspondences by matching local features like SIFT [120], SURF [176] or ORB [127, 177, 178]. Those approaches are not feasible for localizing against the 3D LiDAR map as its lack local visual features. The main difficulty of localizing single image within a LiDAR map is to handle the inherent modal differences between 2D RGB image and 3D point clouds. Recent works can be divided into two categories: matching in 2D space and matching in 3D space. Methods based on 2D matching synthesize images from 3D points based on LiDAR reflectance or distance and compare it with the query RGB images. For instance, Wolcott et al. [179] construct a LiDAR reflectance image database and perform localization under the image retrieval framework. The similarity metric is designed with the normalized mutual information (NMI). Newman et al. [180] propose a method by matching the query images against generated LiDAR intensity images, and they solve the localization problem through a Quasi-Newton optimization. Newber et al. [181] produce a depth image from two images and match it to the intensity image from the LiDAR 3D map. Xu et al. [182] present a method by matching the depth images against the LiDAR reflectance images. Kim et al. [183] synthesize depth images and formulate the cost function with the difference of synthesized depth image and depth images generated from a stereo camera. They also utilize Quasi-Newton optimization to localize the query image. Performing 2D-matching involves a huge number of images rendering on-line or off-line, thus it suffers from poor efficiency issues. Moreover, it is vulnerable to the scene changes.

3D matching-based methods perform localization by exploiting the geometry in 3D space. They generate a sparse point cloud through SfM or bundle adjustment and perform the localization using the 3D point cloud registration approaches [184]. Forster et al. [185] localize the query images by aligning the generated 3D points to a 3D map constructed from a depth sensor. Caseliz et al. [186] use similar strategy and align the 3D sparse structure to a prior 3D map. Bao et al. [187] utilize the stereo camera to

reconstruct the side view of the scene and match it against the map. Methods based on matching in 3D space obtain high accuracy without the large storage requirement to store the images. The main drawback is that it is costly to get the 3D geometry of the scene.

Localization in 2D space cost huge time in rendering 2D images especially for large scenes. Furthermore, comparing the query image with large numbers of rendering images also influences the efficiency. For localization methods in 3D space, most of time is spent on the matching space.

2.5 Summary

In this chapter, we have reviewed related image-based localization methods and the literature in topological localization and metric localization. For topological localization, node representation and matching strategies are reviewed and the limitations are analysed. With regarding to metric localization, three types of methods are reviewed and analysed including image retrieval-based methods, SfM-based method, and deep learning-based methods. Their advantages and disadvantages are presented respectively. We also have reviewed the related methods on image depth prediction and 3D matching, which are important for single image depth prediction.

In next chapter, we will describe our topological localization method with semantic landmark sequences in detail.

CHAPTER 3

Indoor Topological Localization using Semantic Landmarks

This chapter presents a novel indoor topological localization method based on mobile phone videos. Conventional methods suffer from indoor dynamic environmental changes and scene ambiguity. The proposed Visual Landmark Sequence-based Indoor Localization (VLSIL) method is capable of addressing the problems by taking the indoor objects as landmarks. Unlike many feature or appearance matching-based localization methods, our method utilizes highly abstracted landmark semantic information to represent locations, and thus is invariant to illumination changes, temporal variations and occlusions. We match consistently detected landmarks against the topological map based on occurrence order in the videos. The proposed approach contains two components: a convolutional neural network (CNN)-based landmark detector and a topological matching algorithm. The proposed detector is capable of reliably and accurately detecting landmarks. The other part is the matching algorithm built on the second order hidden Markov model and it can successfully handle the environmental ambiguity by fusing semantic and connectivity information of landmarks. To evaluate the method, we conduct extensive experiments on the real world dataset collected in two indoor environments, and the results show that our deep neural network-based indoor landmark detector accurately detects all landmarks and is expected to be utilized in similar environments without retraining, and that VLSIL can effectively localize indoor landmarks.

This chapter is organized as follows. Section 3.1 describes the applications of the topological localization and the advantages of the proposed semantic topological map. In Section 3.2, we illustrate the basic concept of visual landmark sequence-based indoor localization. Section 3.3 presents the detail of CNN-based detector which detects land-

marks from smart-phone videos. Section 3.4 elaborates the proposed matching algorithm based on second order hidden Markov model. Section 3.5 presents extensive experimental results and Section 3.6 concludes the chapter.

3.1 Introduction

Topological localization is one of the fundamental components for pedestrians and robots localization, navigation and mobile mapping [188, 189]. It is compatible with human understanding as topological maps utilize the highly abstracted knowledge to present locations. Represented by a graph, a topological map is a compact and memory-saving approach to represent the environment, and thus is suitable for large scale localization [190]. Each node indicates a region of the environment and is associated with a visual feature vector. The vital problem of the technique is to design robust and distinctive features to represent nodes distinctively.

Many hand-crafted features have been devised based on colours, gradients [190], lines [55] and distinctive points to represent the nodes. Previous work also learn the representation of the nodes using machine learning techniques. However, most of them fail in dynamic indoor environments due to camera noise, illumination and perspective changes and temporal variations. Another serious problem is that there are numbers of visually similar locations in the same environment, which further adds the difficulty of finding the proper visual location representation. Therefore, it still remains a challenging problem for vision-based indoor localization.

Exploiting semantic information from videos for localization is more feasible and human-friendly compared to conventional feature or appearance matching-based methods. Finding matched features in large scenes is inefficient, and it often fails due to the numbers of visually similar locations. Besides, matching multi-modality images is also a problem. Steady elements in the environment are robust representations for locations as they are salient and insensitive to occlusions, illuminations and view variations. In addition, their ground truth locations are fixed and known.

In this thesis, we propose a robust landmark representation using semantic information. A CNN-based landmark detector is proposed to extract landmark semantic information. Unlike previous approaches using hand-crafted features, our detector learns the distinctive features to distinguish target objects and background. Moreover, it can be used off-the-shelf scenes without retraining. The learned features are not derived from a single space but a combination of colour, gradient and geometric space. With proper training dataset, it stays robust to landmarks variations caused by illumination

and other deformations. CNN is selected due to its high performance in image classification [191] and indoor scenes recognition[192] and outperforms approaches based on hand-crafted feature.

Besides, we also propose a novel visual landmark sequence-based approach for indoor topological localization. In the approach, semantic information of steady objects is used to represent locations, and their occurrence orders in the video are used for localization in combination with their semantic information. A topological map constructed with the prior of floor plan map of the environment is used to indicate connectivity information between landmarks. Each node on the map indicates a local region of the environment and is represented by the landmark. To address the environmental ambiguity problem, we extract landmark sequence from a mobile phone video, and match them using the proposed matching algorithm. We make the following original contributions:

1. We propose a novel Visual Landmark Sequence-based Indoor Localization (VLSIL) framework to acquire indoor location through smart-phone videos.
2. We propose a novel topological node representation using semantic information of indoor objects.
3. We present a robust landmark detector using convolutional neural network for landmark detection that does not need to retrain for new environment.
4. We present a novel landmark localization system built on a Second order hidden Markov model to combine landmark semantic and connectivity information for localization, which is shown to relieve the scene ambiguity problem where traditional methods have failed.

3.2 Visual Landmark Sequence-based Indoor Localization (VLSIL)

We propose a novel Visual Landmark Sequence-based Indoor Localization (VLSIL) framework and we first illustrate its basic idea. Suppose there is an indoor space that has 7 locations as shown in Figure 3.1a. For each location, there is a landmark representing it as shown in Figure 3.1b, and the colour indicates the landmark type. Pedestrians can only walk from one location to the others linked by a path. Suppose pedestrians reach the location L(2) without knowing it and observe the red landmark. Their locations can not be determined since there are more than one locations denoted by the red landmark (e.g. LM(5) and LM(7)). Suppose pedestrians observe red, green

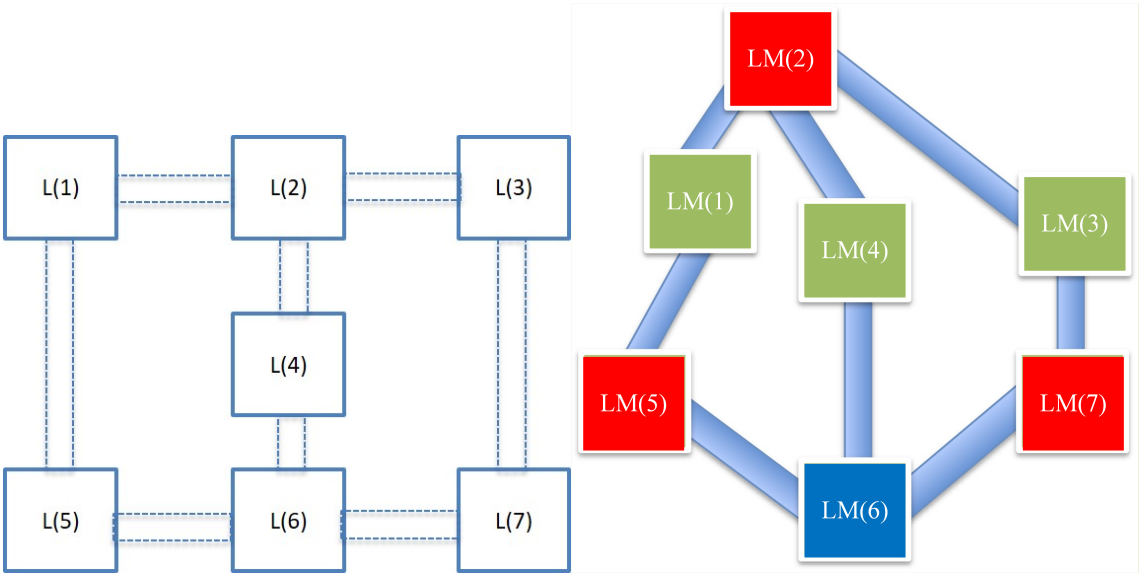


Figure 3.1: (a) Topological map of an indoor space, where there are 7 locations. (b) In each of the locations of the space, there is a landmark representing it. Landmarks of the same colour are identical (e.g. office doors). A person can only walk from one location to the next linked by a path.

and blue landmarks in sequence in their path, then they can be sure they start from LM(2), go through LM(4) and arrive at LM(6), because LM(2), LM(4) and LM(6) are the only valid path. The VLSIL achieves localization through taking photos (video) of a location to determine the current position by matching a sequence of previously discovered landmarks against the topological map of the space.

3.3 Landmark Detection

Landmark detection process consists of two phases: offline phase and online phase. During offline phase, landmark types are pre-defined from the common indoor objects and scenes, and a convolutional neural network is trained to recognize them. Online phase performs the landmark detection from captured videos. It includes frame extraction, region proposal and landmark type determination. Figure 3.2 illustrates the whole process. The offline phase is highlighted with light blue background and the rest is online phase.

In the real scene, majority of the extracted images only capture the background information, which are usually walls. Applying selective search to these images is not necessary and decreases the efficiency. Therefore, we first determine whether the ex-

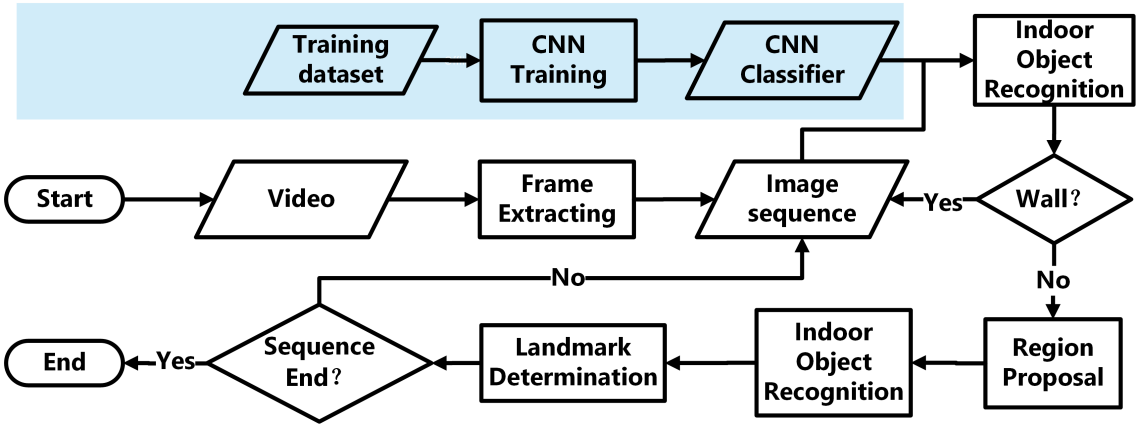


Figure 3.2: Flowchart of indoor landmark detection. It is comprised of two main phases: online phase and offline phase (highlighted with light blue background). The online phase consists of frame extraction, region proposal, indoor object recognition and landmark type determination.

tracted image belongs to wall (background). If so, next image is proceeded. If not, selective search is performed and to find the landmarks.

The rest of section gives a detailed introduction of offline phase and online phase of the process.

3.3.1 Offline Phase

Landmark Definition. Landmarks are defined using common indoor objects like doors, fire extinguishers and stairs, and indoor structure locations. Some examples of common objects are shown in Figure 3.3. Other indoor objects like chairs and desks are not used because their positions are not fixed.

Three types of landmarks are defined : single object landmarks, multiple-object landmarks, and scene landmarks. Single object landmarks consist of one object such as a fire extinguisher or an elevator. multiple-object landmarks are defined with more than one objects. For instance, office doors are multiple-object landmarks, which include a doorplate and a door. Combining multiple objects enlarges the landmark distinctiveness and reduces ambiguity of the map. We do not utilize the texts in the doorplate to further distinguish the office doors because motion blur makes the text recognition very challenging. Scene landmarks are key locations of the indoor structure such as corners, intersections or halls that have unique visual patterns.

Training CNN-based indoor object classifier. Our landmark detection relies on the object detection results of the extracted images. High accuracy and real-time performance of CNN on object detection inspires us to choose it for our application [193]. In the ap-

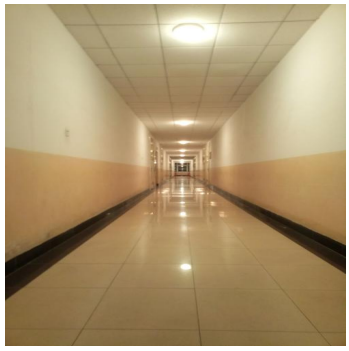
(a) Fire extinguisher

(b) Stair

(c) Door

(d) Elevator

(e) Toilets tag



(f) Intersection

Figure 3.3: Common indoor objects and locations of interest.

plication, we develop our CNN-based landmark detector by modifying AlexNet [194]. The modified AlexNet contains 5 convolutional layers and two fully connected layers. Each convolutional layer is tailed by a max pooling layer. Two fully connected layers are used to assemble information from the convolutional layers. AlexNet is selected for two reasons. The first is that it has proved its high performance in image classification in ImageNet competition. Secondly, it is relatively easy to converge since it has relatively fewer layers compared to other more complex networks.

Several tricks are applied to train AlexNet for our indoor object detection. Firstly, the output layer has to be adjusted to recognize the target indoor objects. AlexNet is originally designed for ImageNet competition, which aims to recognize 1000 types of objects. However, not all indoor objects of our interest are included. We replace output layer with new one, in which the number of neurons equals to the number of our interesting indoor objects. Softmax function is chosen as the activation function of output layer neurons. Secondly, we retrain AlexNet with fine-tuning technique. Only the newly added layer is allowed to retrain, while the weights of the rest of the layers are fixed. Finally, to eliminate the object variations caused by illuminations, rotations and movement, we conduct data augmentation by pre-processing original images. For each

original image, we change its brightness by adding 10, 30, -10, and -30 to produce new images. We rotate original image by 5°, 10°, -5° and -10°. The movement of pedestrians leads to the partial occlusion of targets of interest. We also generate new image by randomly cropping original image into size of 224×224 pixel. Altering the brightness and rotating images are done with the original images and cropping step is done during the training stage. In this way, we enlarge the training dataset and the trained network is robust to those variations.

3.3.2 Online Phase

The online phase consists of frame extraction, region proposal, indoor object recognition, and landmark type determination. We elaborate the procedures in detail, except the indoor object recognition step which simply feeds the image patches into the classifier.

Frame Extraction. During the online phase, smart-phone videos are sampled at a given rate. Sampling rate is a vital parameter as it impairs landmark detection accuracy and efficiency. Low sampling rate results in low overlap or even no overlap between successive images, which leads to the lost of track of certain objects in the image sequence. High sampling rate leads to large information redundancy, resulting in low landmark detection efficiency as more images are to be processed. Overlap can be roughly estimated using equations (3.3.1) and (3.3.2). They are applied in two scenarios: walking along a line and turning to another direction.

$$Overlap = 1 - \frac{V}{2H\tan(\frac{\theta}{2})Hz} \times 100\% \tag{3.3.1}$$

$$Overlap = 1 - \frac{V_{ang}}{Hz\theta} \times 100\% \tag{3.3.2}$$

where V represents walking speed and H is the average distance between camera and surrounding environment. θ is the field of view of camera in each mobile phone. Hz represents sampling rate. V_{ang} is the angular velocity. Empirically, the sampling rate of 3-5 frames per second would work well according to the general walking speed of human beings.

Region Proposal. Cutting target objects out of extracted images is crucial for landmark detection. Feeding images that contain background, and target objects directly into the classifier decreases the object recognition accuracy. It is because training samples are covered with indoor objects in the majority of image space, while in extracted images,

target objects may occupy only a small part of the extracted image. Therefore, we have to crop the patches with target objects taking up most of the space. Here we choose the selective search algorithm to generate patches of interest from images [195]. Selective search employs a bottom-up strategy to generate patches. The process contains two steps. At first, an over-segmentation algorithm is applied to generate massive initial regions in a variety of colour space with a range of different parameters. Then a hierarchical grouping approach is performed based on diverse similarity measurements including colour, texture, shape and fill, with various starting points. Hundreds or thousands of patches are produced from this algorithm. However, we do not need to process all of them to identify the target objects since eligible patches may be too many. Normally, the selective search generates thousands of patches using default parameters, and from these we randomly chose 300 patches for accuracy and efficiency reasons.

Landmark Type Determination. Landmark type is determined based on the indoor objects recognition results. For single object landmarks and scene landmarks, their types are given with their corresponding indoor objects. For example, if an elevator is detected, an elevator landmark is detected. Regarding to multiple-object landmarks, their types are determined when their components are correctly detected. For instance, if the doorplate and door are detected in the same image image or a short image sequence, then an office door landmark is detected.

A sequence of images are used to perform landmark type determination instead of a single image. The main reason is that components of multiple objects landmarks might not appear in the same image. The recognition result of a sequence images can address the problem as the components are sequentially detected. Besides, it is helpful to eliminate the wrong recognition results. In this thesis, indoor objects that are not seen in 3 successive images is taken as the false detection. Exploiting images sequence for localization also helps determine the landmark occurrence order when more than one landmarks are observed in a single image.The first landmark detected prior to the current landmark is viewed as the previous landmark of the current detected landmark in the sequence. Sequence image length is set automatically based on the recognition results. A sequence starts from an object is robust recognized and ends at the images that are walls.

3.4 Visual Landmark Sequence Localization using Second Order Hidden Markov Model

Knowing a sequence of landmark types from a video, we match them with the predefined topological map. In this section, we illustrate the defined topological map and the matching algorithm based on Second Order Hidden Markov Model for our applications. We also extend the Viterbi algorithm for our application.

3.4.1 Topological Map

The topological map provides the information of the distribution of landmarks of the indoor environment and indicates the connectivities between landmarks. In our case, topological map is a directed graph, and is created from the floor plan map of the indoor environment. It consists of two types of elements: nodes and edges. Nodes indicate regions of the environment. Its colour represents landmark type. In this thesis, we use red nodes for fire extinguishers, black for intersections, blue for offices, silver for elevators, yellow for stairs, light green for the disabled toilets, green for man's toilets and dark green for Woman's toilets. Edges denote the connecting information between landmarks. An edge starting from node i to node j indicates the sequential direction that landmark j is detected after landmark i. Arrowed line indicates one way connection. In certain situation, two landmarks might be spatially close to each other. They are viewed as two regions, and are represented with the corresponding landmarks.

3.4.2 Second Order Hidden Markov Model for Indoor Localization

Second order hidden Markov model (HMM2) takes context information to perform tasks. It contains 5 elements: observations set, states set, initial probability, emission matrix and transition matrix. For our application, observations set includes all landmark type and states set indicates the landmark locations. Initial probability represents the starting position of a route. In the rest of the section, we detail the emission matrix and transition matrix of HMM2 in our scenario. We also introduce a new parameter to handle unidentified multiple objects landmarks.

Emission Matrix of HMM2. Emission matrix represents the state probabilistic distribution over observation set [196]. Its row count equals to the number of states and its column count is the number of the observations classes. For our problem, the entry values of emission matrix indicate the probability of an observed landmark type, which belong to a certain state. We assign the emission matrix value based on landmark types

of a landmark location. The emission matrix is defined as follows: $e_{i,j} = 1$, if landmark type j corresponds to state i; $e_{i,j} = 0$, otherwise.

Transition Matrix of HMM2. Unlike transition matrix of hidden Markov model which is a 2-dimensional matrix, the transition matrix of HMM2 is 3-dimensional [196]. Its value $t_{i,j,k}$ indicates the probability that next state is k, given the condition that previous state is i and current state is j. For landmark-based indoor localization problem, it represents probability of going through certain landmark position given previous two landmarks positions. The matrix is defined as: $t_{i,j,k} = 1$, if there is a path from i through j to k; $t_{i,j,k} = 0$, otherwise.

Probabilistic Matrix of Landmark Type. Ideally, multiple object landmark type should be correctly recognized. But in some cases, only a component of the landmark is detected for various reasons. To deal with the problem, a probabilistic matrix, $p_{i,j}$, the probability of landmark type i given detected object j, is defined. This parameter does not affect single object landmark and scene landmark. For them, when the object or scene is detected, its landmark type is determined. It aims to solve the confusion of multiple objects landmark when part of landmark is observed. It works for the situations where an object is detected but its landmark type still remains undetermined. The matrix value $p_{i,j} = 1$, if landmark i is a single object landmark and j is the object to form it, $p_{i,j} = 0$, otherwise. For multiple objects landmark, if the detected object is not able to be used to recognize landmark, we split the probability evenly. For example, if a door is detected, its matrix value equals to 0.25 since it could belong to either an office or a toilet.

3.4.3 Extended Viterbi Algorithm for Indoor Localization

Given modified HMM2 for landmark localization, we extend Viterbi algorithm to find the landmarks sequence corresponding to the sequence of landmark types using Bayesian theory. The details are as below. Assume that the HMM2 has M states for landmarks, and the initial state parameter is π_i, which represents the probability when the process starts from landmark i. Transition matrix value t_{ij} is the transiting probability that the process move from landmark i to landmark$j$. There are n detected landmarks in the observation sequence, represented by $Y = \{y_1, y_2 \ldots y_n\}$. The corresponding locations are represented by $X = \{x_1, x_2 \ldots x_n\}$. We aim to find the landmark location sequence X of the maximum probability, given the landmark type sequence Y. Therefore, our objective function is to maximize $P(X|Y)$. From Bayesian theory,

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{3.4.1}$$

where $P(Y|X)$ denotes the probability distribution of landmark type sequence Y, given state sequence X. In Hidden Markov Model(HMM), it is represented by emission matrix. $P(X)$ is the prior probability distribution of state sequence X. $P(Y)$ is the probability distribution of observation sequence. It is a constant value. Hence the solution to maximizing $P(X|Y)$ and maximizing $gu(X)$ are the same.

$$gu(X) = P(Y|X)P(X) \tag{3.4.2}$$

Taking logarithm of $gu(X)$, equation (3.4.2) is changed to equation (3.4.3).

$$lgu(X) = log(gu(X)) = \sum_{j=1}^{n} logP(y_i|x_i) + logP(x_1, x_2, \ldots\ x_n) \tag{3.4.3}$$

Since logarithm function is monotonically increasing, $lgu(X)$ and $gu(X)$ share the same solution for the maximization problem. Note that HMM requires that the next state only depends on the current state. $LogP(x_1, x_2\ \ldots\ , x_n)$ can be simplified to equation (3.4.4).

$$logP(x_1, x_2, \ldots\ x_n) = (\sum_{j=2}^{n} logP(x_j|x_{j-1})) + logP(x_1) \tag{3.4.4}$$

Equation (3.4.3) is transformed to equation (3.4.5).

$$lgu(X) = \sum_{j=1}^{n} logP(y_i|x_i) + (\sum_{j=2}^{n} logP(x_j|x_{j-1})) + logP(x_1) \tag{3.4.5}$$

The Viterbi algorithm is used to find solution to the maximization of $lgu(X)$. It recursively computes the path. Two parameters are updated in the process. At any step t, $V_{t,k}$ is used to record the maximum probability of the landmarks sequence ending at landmark k , given t observations. $Ptr(k,t)$ records the previous landmarks before landmark k in the most likely state sequence. The process is as follows.

$$V_{1,k} = e_{y_1,k} \times \pi_k \tag{3.4.6}$$

$$V_{t,k} = max(e_{y_t,k} \times t_{x_{t-1},k} \times V_{t-1,x_{t-1}}) \tag{3.4.7}$$

$$Ptr(k,t) = \arg\max_{k}(e_{y_t,k} \times t_{x_{t-1},k} \times V_{t-1,x_{t-1}}) \tag{3.4.8}$$

Viterbi algorithm has shown its good performance in solving HMM problem. It has to be modified to solve HMM2 problem because HMM2 takes both the previous state and the current state into consideration when predicting next step. Thus equation (3.4.4) has to be extended as follows.

$$logP(x_1, \ldots, x_n) = \sum_{j=3}^{n} logP(x_j|x_{j-1}, x_{j-2}) + logP(x_2|x_1) + logP(x_1) \tag{3.4.9}$$

Another issue is that during landmark detection the landmark type might not be clearly recognized. The modified equation is equation (3.4.9). A parameter is added to represent such unclear observation. The Viterbi algorithm for HMM2 is initialized by equations (3.4.10) and (3.4.11) followed by iteration equations (3.4.12) and (3.4.13), and is summarized in Algorithm 1.

$$V_{1,k} = max(p_{y_1,s_1} \times e_{y_1,k} \times \pi_k) \tag{3.4.10}$$

$$V_2(x_1, k) = V_{1,x_1} \times t_1(x_1, k) \times max(p_{y_2,s_2}) \times e_{y_2,k} \tag{3.4.11}$$

$$V_t(x_{t-1}, k) = max(V_{t-1}(x_{t-2}, x_{t-1}) \times t_2(x_{t-2}, x_{t-1}, k)) \times max(p_{y_t,s_t} \times e_{y_t,k}) \tag{3.4.12}$$

$$Ptr_t(x_{t-1}, k) = \arg\max_{x_{t-2}} (V_{t-1}(x_{t-2}, x_{t-1}) \times t_2(x_{t-2}, x_{t-1}, k)) \tag{3.4.13}$$

Where S_t is the object type of detected landmark t.

Algorithm 1: Extended Viterbi finds the location sequence of maximum probability

Input: A sequence of observations Y, transition Matrix T_1, T_2, emission matrix E, probabilistic matrix P initial location π

Output: A sequence of States X

Def: N, number of locations; M, number of landmark type; n, number of observations

Initialization:

  $V_1 = T_1 \times \pi \times E \times P$

Recursion:

  $V_t = V_{t-1} \times T_2 \times E_t \times P_t$

  $Ptr_t = \arg\max (V_{t-1} \times T_2)$

Back trace:

  $X_K = \arg\max_{col} (V_N)$ column index of the V

  $X_{K-1} = \arg\max_{row} (V_N)$ row index of the V

  $X_t = Ptr_{t+1}(X_{t+1}, X_{t+2})$

Return X;

3.5 Evaluation

3.5.1 Setup

To evaluate the proposed method, we conducted our experiments on B floor of the Business South building (BSB) and B floor of the School of Computer Science building (CSB) at the University of Nottingham, UK. The two sites are typical office environments containing many corridors and office rooms. Floor plan maps of two sites are shown in Figure 3.4 and Figure 3.5 respectively, and their corresponding topological maps are shown in Figure 3.6 and Figure 3.7. We selected eight types of landmarks from the two places: office room, stair, elevator, fire extinguisher, men's toilet, women's toilet, disabled toilet and intersection (corner). Among them, fire extinguisher, stair and elevator belong to single-object landmarks. Office rooms and toilets are multiple-object landmarks. Intersection is scene landmark. BSB is a relatively simple environment while CSB building is more complex. In the School of the BSB, there are 54 landmarks in total, and 65 landmarks in the CSB.

Two female and three male participants were asked to collect videos in both sites using smart-phones. Three models of mobile phones were used: an Huawei Honor, a Samsung Note 3 and an iPhone 6s Plus. Each participant wore a mobile phone on their upper arm, with the camera looking sideways. The participants are required to walk forward smoothly without standing still. They also should not shake their arms while walking to avoid the image blur caused by rapid motion. Taking side-viewed videos provides more information about landmarks as it is orthographic projection on landmarks. Compared to front-view, view variations are relieved. Another reason is that side-view capturing has a narrow field of view, which facilitates the determination of the landmark occurrence order, since the landmarks appear one by one in the video. Participants were asked to walk freely along the corridors in two experimental sites. In our experiments, a real world mobile video dataset of 1.9 hours in total was collected for the evaluation of the proposed method. The videos from the CSB are collected in the morning of the weekends; participants walked at normal speed. The videos from BSB are captured in the afternoon. The walking speed was about 1.5 metres per second on average.

Seven routes were used as testingbed to evaluate our method. Two of them were collected in the BSB and five of them from CSB. Route 1-2 are from the BSB and Route 3-7 are from CSB. The overview of the 7 routes are as follows:

Route 1: The route begins at node 43 and goes through 28 landmarks, ending at node 47.

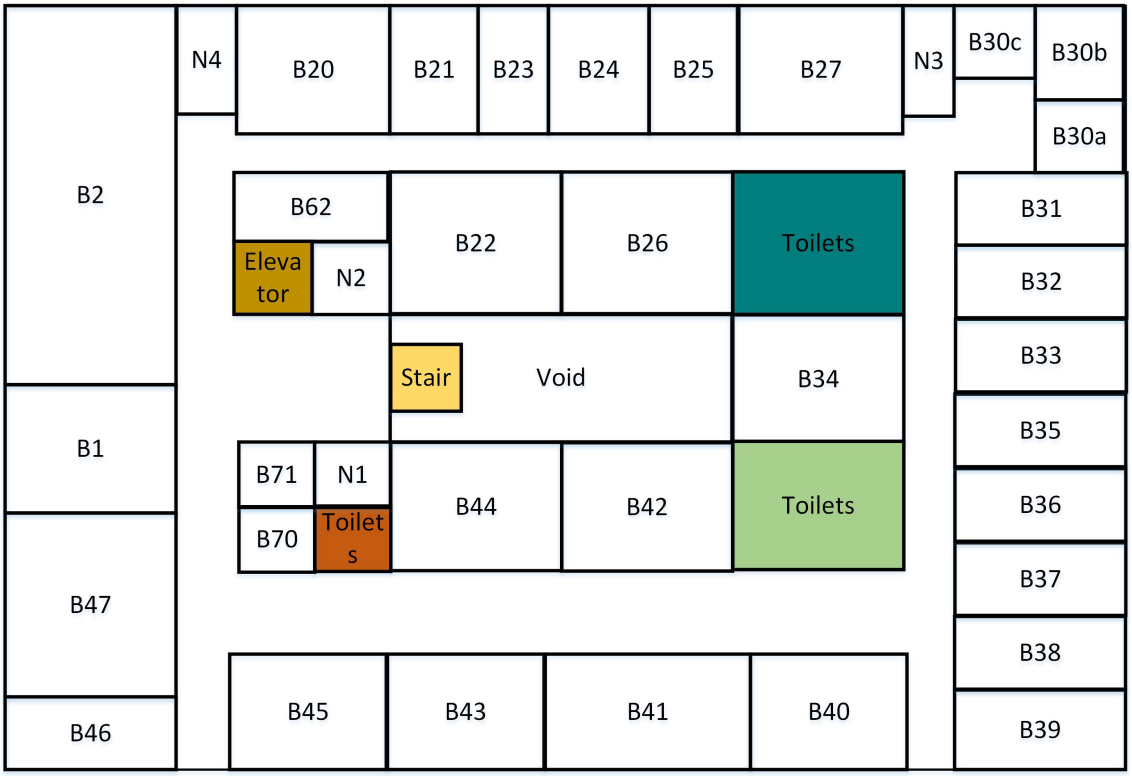


Figure 3.4: Floor plan map of B floor in the BSB.

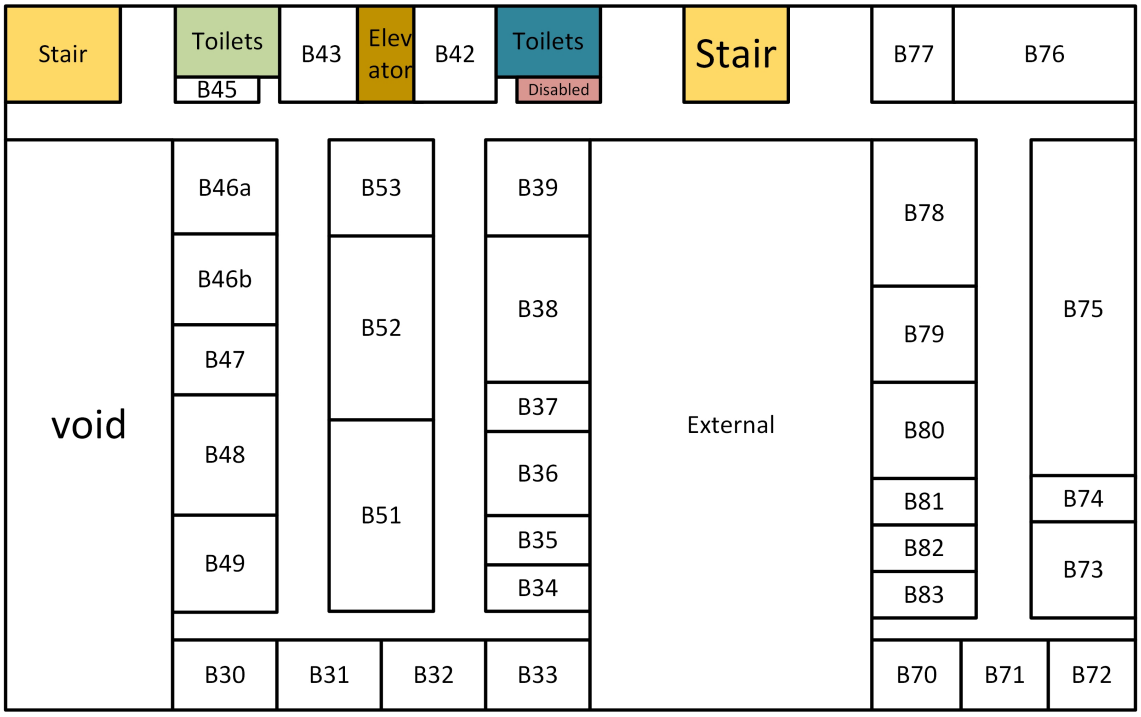


Figure 3.5: Floor plan map of B floor in the CSB.

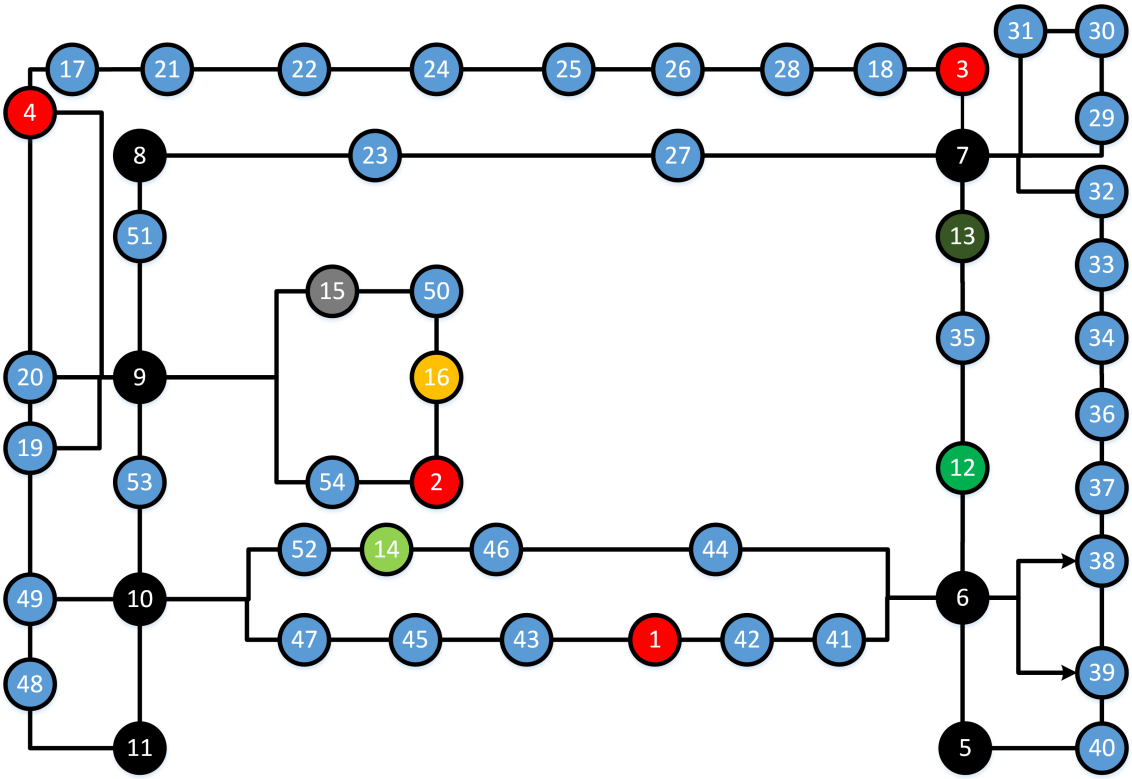


Figure 3.6: Landmark topological map of B floor in the BSB.

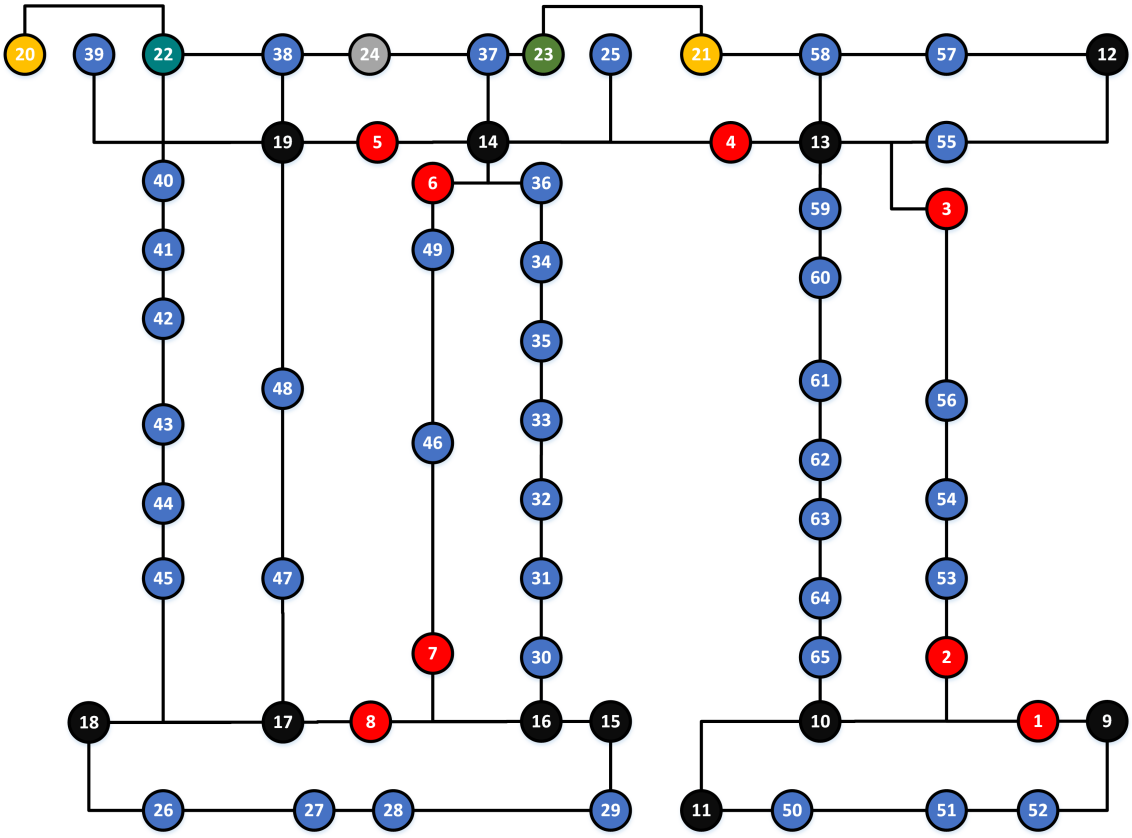


Figure 3.7: Landmark topological map of B floor in the CSB.

Route 2: The route starts from the node 44 and turns left at all four turns before ending at node 46. There are 16 landmarks in this route.

Route 3: This route goes through 15 landmarks. It starts from an office door (node 52) and ends in the intersection (node 14). It walks through a sequence of office door, containing a corner and a left turn.

Route 4: The route starts from the left stair and goes straight to the end corner of the corridor. In total, 10 landmarks are included in this route.

Route 5: This route contains 14 landmarks. It begins from an intersection (node 16) and goes through a sequence of office doors, turn, elevator and finally reaches the left stair.

Route 6: This route starts from a turn (node 16) and ends at an office (node 65) , going through 3 turns, containing 17 landmarks.

Route 7: The route begins from a turn named node 16 and goes to the end of the corner before turning left. It goes straight until reaching the turn (node 19). It goes down to the turn (node 17). There are 22 landmarks in this route.

3.5.2 Landmark Detection

Indoor Objects Recognition. The selected landmarks are comprised of nine classes of indoor objects, including eight classes of indoor objects: door (DR), women's toilet tag (WMTT), men's toilet tag (MTT), disabled toilet tag (DTT), fire extinguisher (FE), door plate (DP), elevator (ELV), and stair (ST) and one class of scene object (corner or intersection)(CN). Together, they form 8 types of landmarks. We also introduce background as a type class during training process, which are uninteresting object (walls mostly (WLL)). Uninteresting objects act as negative training samples. This increases the determinativeness and generalization ability of the classifier.

We collected about 1300 images containing these ten types of indoor objects (nine of them are objects of interest and one is background). About 1000 of them were used for training (fine-tuning the CNN pre-trained on ImageNet data) and the rest for testing. The distribution of training and testing dataset are shown in Table 3.1 These data came from two sources, images on the Internet and video frames of collected data. We leveraged images from the Internet for two reasons. Firstly, the training dataset could be enlarged, and thus the discriminative capacity of trained classifier over targeted indoor object is improved. Another reason is that our detector can be used in a new environment without retraining.

We selected AlexNet as the basic network and fine-tuned it for our application. The

Table 3.1: Distribution of training and testing data.

Type	CN	DTT	DR	DP	ELV	FE	MTT	ST	WLL	WMTT
Training	56	60	155	63	60	250	58	113	104	55
Testing	29	25	33	22	23	36	24	37	31	20

output layer was modified by changing the number of neurons from 1000 to 10. Its parameters were initialized with a normal Gaussian distribution. The other layers were initialized with weights that won Visual Recognition Challenge in 2012. Parameters of the convolutional layers and fully connected layers were kept fixed and only the parameter of output layer were learned during the training phase. The CNN network was implemented using the Caffe framework [197]. The learning rate was 0.05 and the maximum iteration was 40000. The network was trained in an MSI laptop in GPU mode. The laptop features a Windows 10 operating system and the processor is Intel i7, and the laptop is fitted with 8GB of RAM. The graphics processing unit is an Nvidia GTX970M.

We further compare the proposed the landmark detection method with traditional hand-crafted feature-based methods. Gist feature [32] is used to represent the visual objects and the objects are recognized using the SVM-based and ANN-based methods. We report the result with the accuracy and the F1 value. F1 value is a measure of classification accuracy, which takes both precision and recall into consideration. Precision represents the number of correct classification results divided by all positive results returned by the classifier. Recall is the number of correct results divided by all the ground true positive samples. The F1 value ranges from 0 to 1, and the higher the value is, the better the performance. F1 can be computed with equation (3.5.1).

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{3.5.1}$$

The comparison results are shown in Table 3.3 and in Table 3.2 respectively.

Table 3.2: Performance comparison on indoor objects recognition in terms of accuracy.

Methods	CN	DTT	DR	DP	ELV	FE	MTT	ST	WLL	WMTT	Overall
SVM	17.2%	64.0%	90.9%	68.2%	0.0 %	100%	0.0%	56.8%	3.2%	0.0 %	44.3%
ANN	82.8%	80.0%	97.0%	86.4%	73.9%	97.2%	87.5%	70.3%	61.3%	80.0%	81.8%
Ours	100%	96.0%	100%	95.5%	95.7%	100%	100%	100%	100%	95.0%	98.6%

The results show that our method achieves best results compared to SVM-based and ANN-based methods on both average accuracy and F1 value. For each type of objects,

Table 3.3: Performance comparison on indoor objects recognition in terms of the F1 value.

Methods	CN	DTT	DR	DP	ELV	FE	MTT	ST	WLL	WMTT	AVERAGE
SVM	0.29	0.78	0.50	0.77	Nan	0.44	Nan	0.67	0.06	Nan	Nan
ANN	0.89	0.87	0.84	0.90	0.76	0.77	0.88	0.78	0.70	0.86	0.82
Ours	1	0.96	1	0.98	0.98	1	1	1	0.98	0.93	0.98

our method outperforms the other two on accuracy and F1 value. SVM-based method fails to recognize the elevator and toilets tags. ANN-based method also obtains high accuracy but it tends to classify the wall into other objects. This affects the localization application as it adds non-existing landmarks to the sequence.

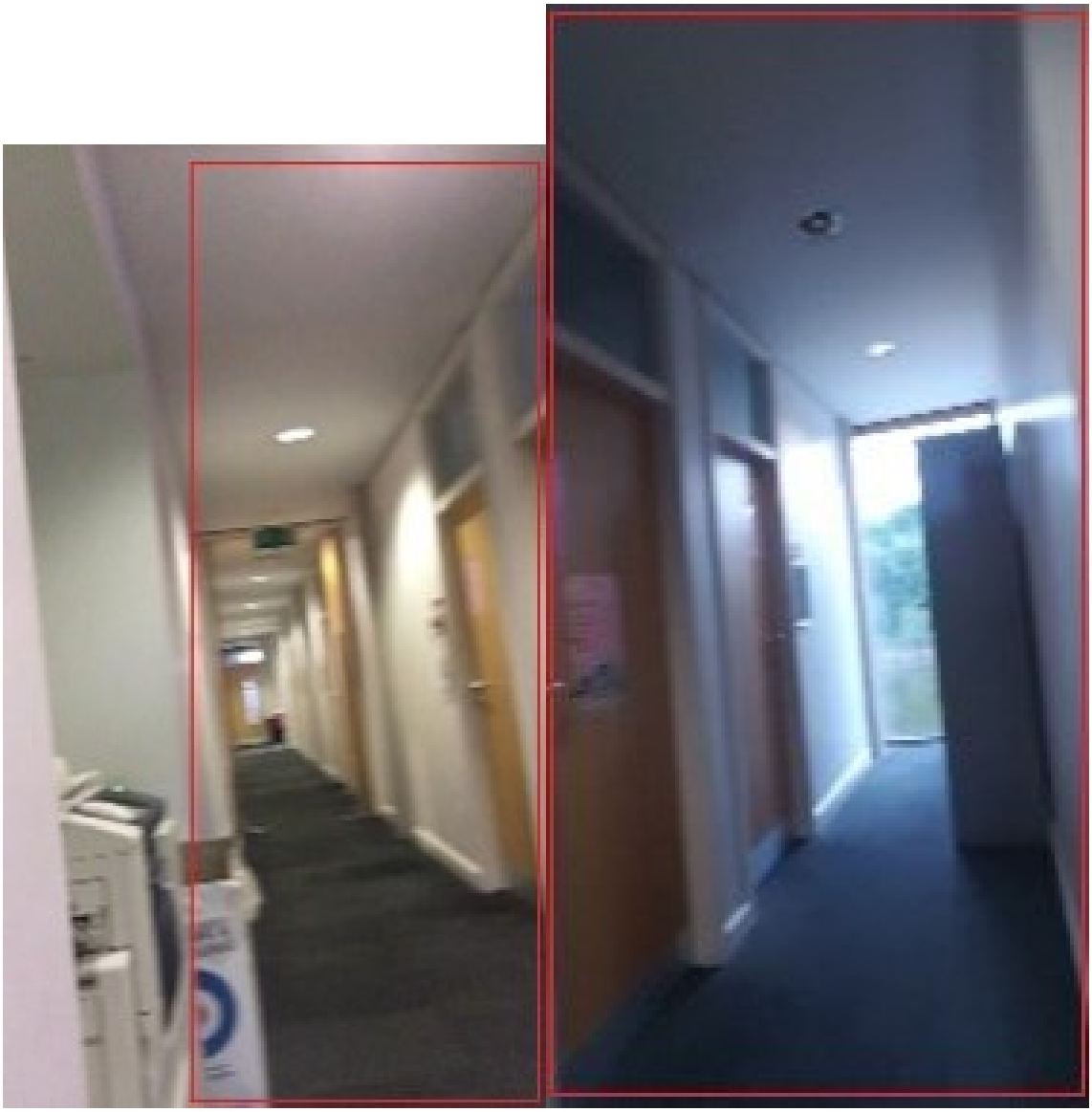
Some cases are shown in Figure 3.8a, 3.8bFigure 3.9 and Figure 3.10 with regarding to the illumination and view change as well as the blur image. It can be seen from that the trained landmark detector is capable of staying robust to the illumination and view change as well as the image blur.

Landmark Detection Performance. All videos of seven routes were empirically sampled at the rate of three frames per second. Seven visual landmark sequences are shown in Figures 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, and 3.17. Sampled images were processed with the selective search algorithm to generate 300 patches. Landmarks were determined from the classification results according to the strategy described in Section 3.3.2.

We applied this trained detector and ANN-based detector to the landmark detection on the 1.9 hours indoor mobile phone videos. SVM-based detector is not used due to its low performance on objects detection. The results are shown in Table 3.4, DL represents detected landmarks, CDL represents the correctly detected landmarks, WDL denotes wrongly detected landmarks.

Table 3.4: Landmark detection performance in the real data test.

Route	Landmarks Counts	ANN			Ours		
		DL	CDL	WDL	DL	CDL	WDL
1	28	30	25	5	28	28	0
2	16	16	16	0	16	16	0
3	15	20	15	5	15	15	0
4	10	10	10	0	10	10	0
5	14	18	14	4	14	14	0
6	18	26	18	6	18	18	0
7	22	29	22	7	22	22	0



(a) high illumination image **(b)** low illumination image

Figure 3.8: Object detection of different illuminations.

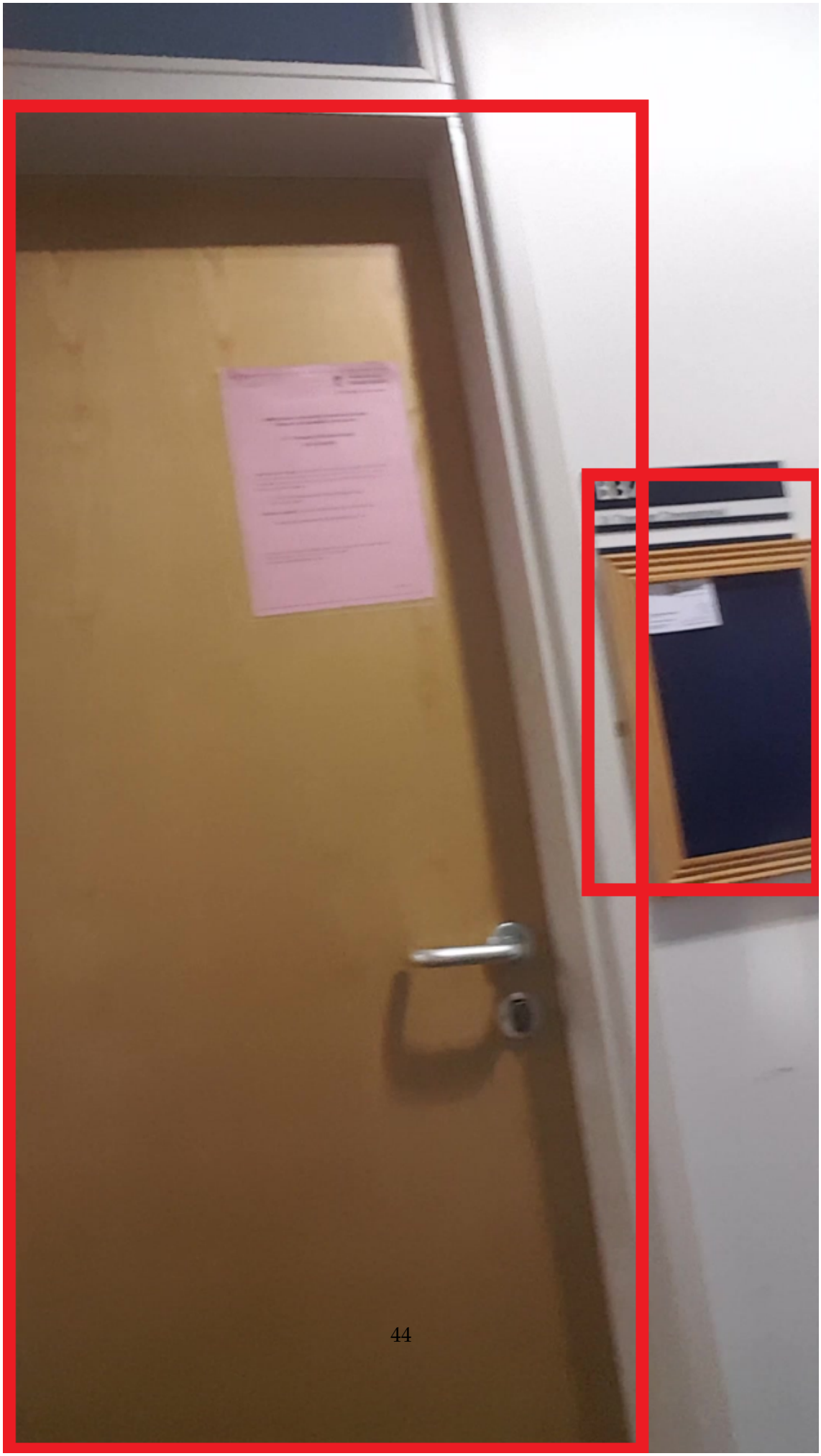


Figure 3.9: Object detection result of the blur images.

Figure 3.10: Object detection result of doors with different views.

Figure 3.11: Landmark sequence of Route 1.

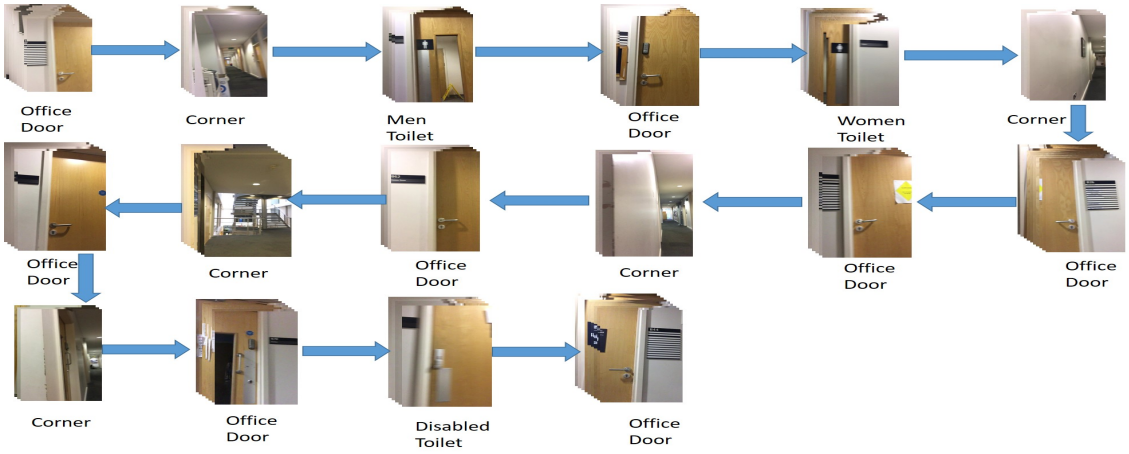


Figure 3.12: Landmark sequence of Route 2.

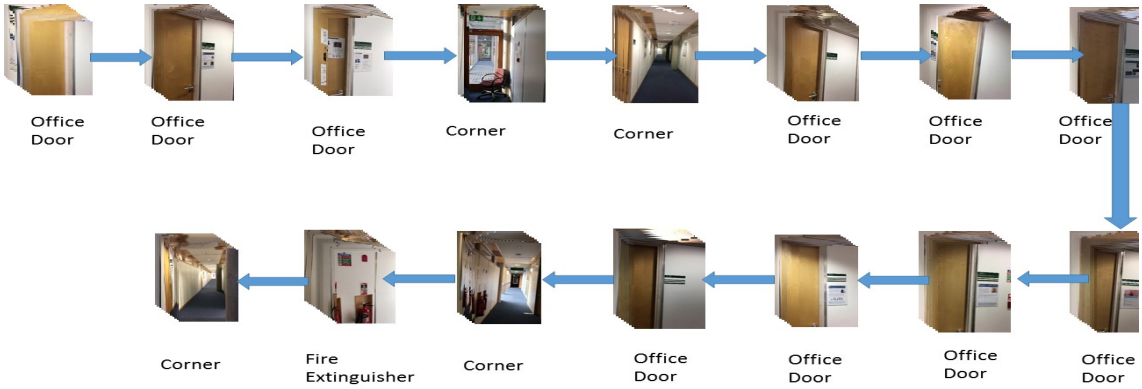


Figure 3.13: Landmark sequence of Route 3.

Figure 3.14: Landmark sequence of Route 4.

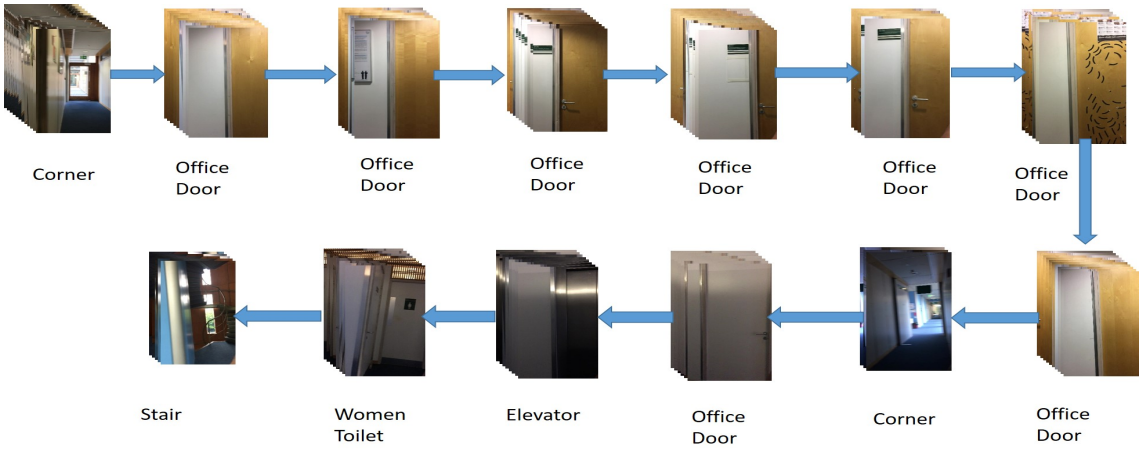


Figure 3.15: Landmark sequence of Route 5.

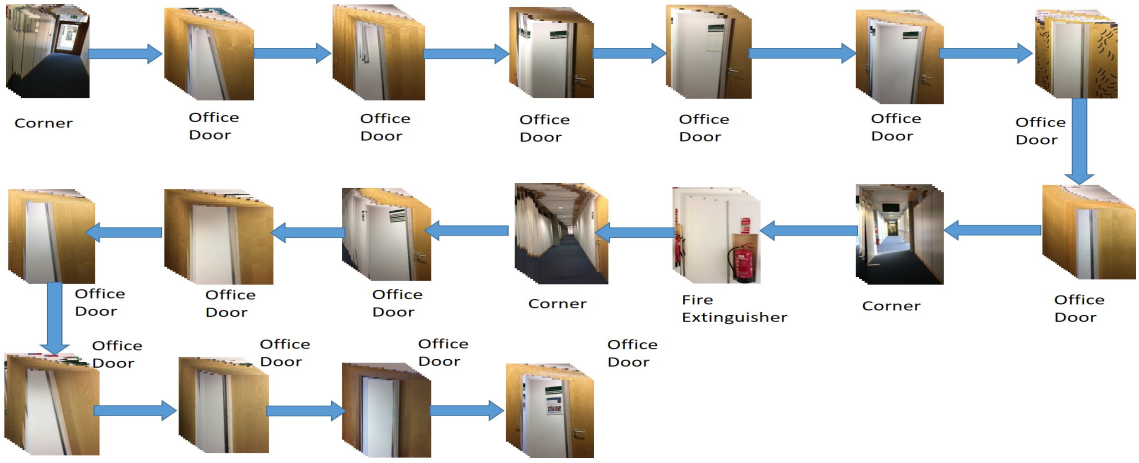


Figure 3.16: Landmark sequence of Route 6.

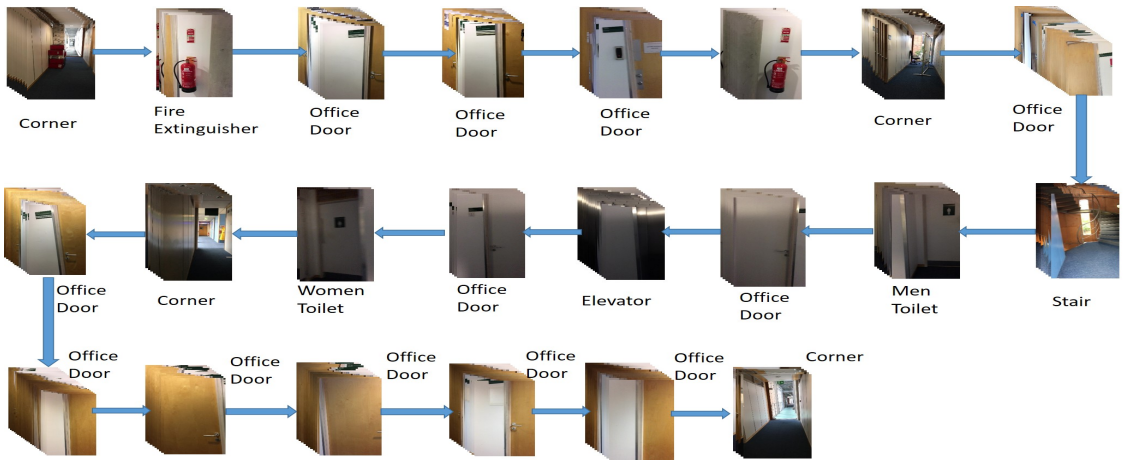


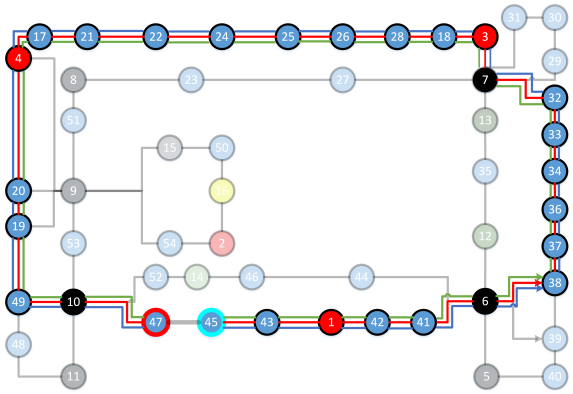
Figure 3.17: Landmark sequence of Route 7.

Our method correctly detected all landmarks in all routes. ANN-based detector correctly detected landmarks in Route 2 and Route 3. Some walls are wrongly detected as doors in Route 3, 5, 6 and 7. This demonstrates that our detector outperforms the detector using hand-crafted feature. Currently, the proposed method can not be achieved in real time. The majority of time are spent on landmark detection. Although the average time of classifying an image is short using our convolutional neural network (about 0.012s on our machine), the average time to process a landmark image is about 7 seconds. The process is time-consuming for two reasons. Firstly, we choose the effective selective search algorithm to generate patches from landmark images, which costs about 3 to 4 seconds to generate reliable patches. Secondly, we feed the 300 patches of a landmark image to the network to correctly detect landmark, which takes up extra 3 seconds. It should be noted that the detection process can be optimized with the development of object detection technologies in computer vision.

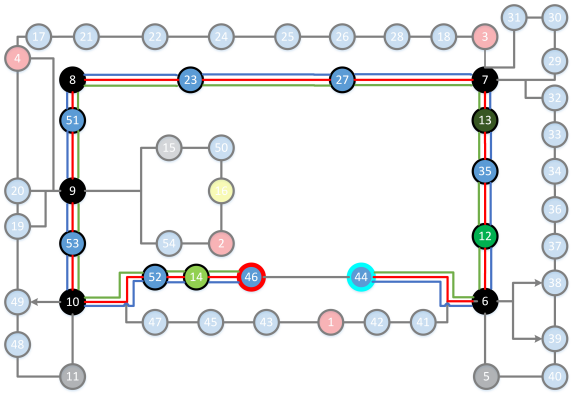
3.5.3 Localization

Performance. We match the detected landmarks with topological map on two situations: with known start and with unknown start. The ground truth routes and the predicted routes are shown in Figure 3.18. The red line indicates the ground truth trajectory. The green line represents the predicted trajectory with unknown start while the blue line represents the predicted trajectory with known start. The route start is represented with node of cyan edge and the route end is denoted as node of red edge.

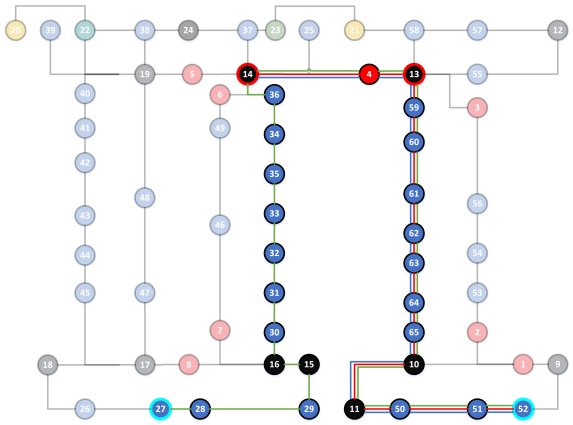
For Route 1, 2, 4, 5 and 7, predictions of both known and unknown start are correctly localized since the blue and green line are in accordance with the red line. For Route 3 and 6, the two blue lines are in accordance with the red lines, indicating that they are accurately localized under known start condition. For unknown start case, Route 3 has two predictions: one starts from node 27 and ends at node 13 and the other one starts from node 52 and ends at node 14. The latter is the correct path. Route 6 also has two predictions: one starts from node 10 and ends at node 30, and the other one begins at node 16 and stops at node 65, the latter of which is correct. This shows that the two routes can not be localized with current observations and further observations are required to be localized eventually. This problem can be solved by giving the start positions since all 7 routes are correctly localized under known start condition. The results demonstrate that our method is capable of localizing users accurately with known start and it also works well in some cases with unknown start. Compared to the landmark detection, the localization process barely costs time. We spend about 0.043 second in average to localize each route.



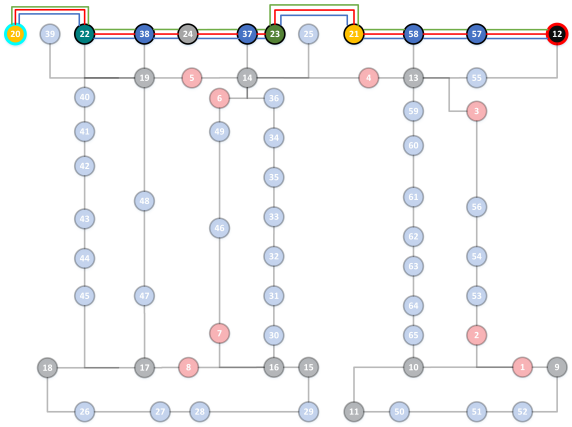
(a) Route 1



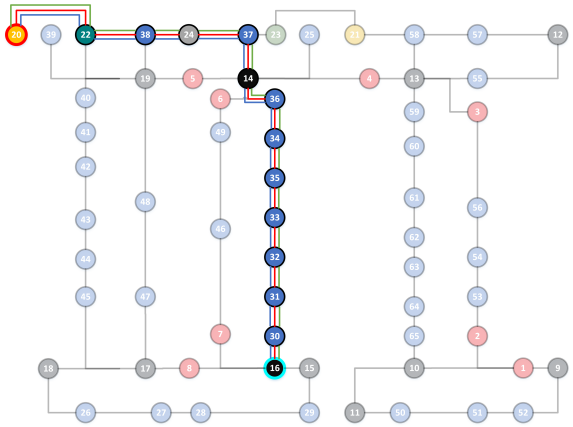
(b) Route 2



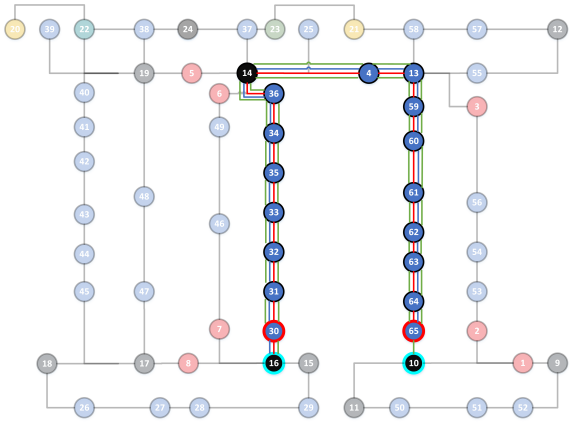
(c) Route 3



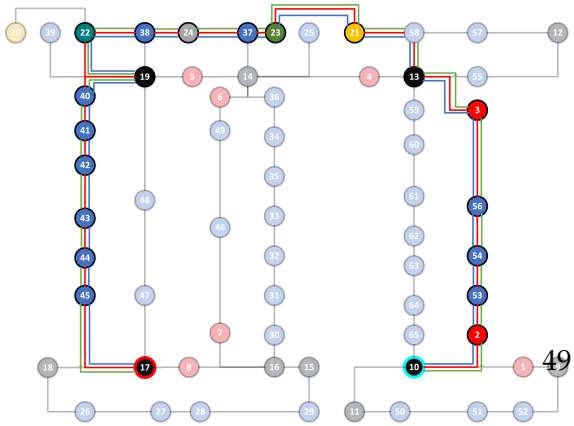
(d) Route 4



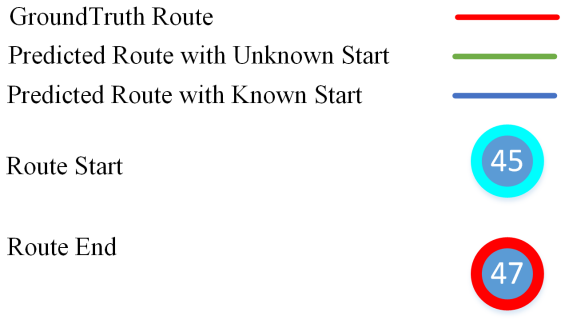
(e) Route 5



(f) Route 6



(g) Route 7



(h)

Figure 3.18: The localization results of 7 routes.

We further make comparison with HMM-based method in two situations and the statistical results are shown in Table 3.5. The number of possible paths is used to report the comparing result. It is notable that HMM fails to localize all landmark sequences without known start and only Route 5 is accurately localized given start position. Besides, our method outperforms the HMM-based method in 7 routes with the same conditions.

Table 3.5: Statistical comparison landmark sequence localization results of 7 Routes.

Route	HMM		HMM2	
	Without	With	Without	With
1	18	9	1	1
2	8	2	1	1
3	1137	82	2	1
4	2	2	1	1
5	12	1	1	1
6	18346	5556	2	1
7	4	2	1	1

Offline performance. Offline matching is done after all the landmarks are detected, and we match the whole landmark sequence with the topological map. The ground truth routes and the predicted routes are shown in Table 3.5. The result shows that the proposed method is capable of localizing users accurately except for Routes 3 and 6 with unknown starting position. This happens for the same reason: the environment is of high ambiguity. This problem can be solved when more landmarks are observed.

Analysis. In this section, we evaluate localization performance of the proposed method regarding to the number of observed landmarks. The number of possible paths are used to report performance. We perform experiments in two scenes using Route 1 and Route 7 along with the number of the observed landmarks under unknown start conditions. The performance is shown in Figure 3.19. It is shown that Route 1 is localized with 6 landmarks and Route 7 is localized at 9th landmark. It is because Computer Science building is more complex compared with the Business South building.

We also conduct experiments to analyse the effects of given route start regarding to the number of observed landmarks. Route 1 from Business South building and Route 3, 7 from Computer Science building are used to perform experiments. It can be seen from the Figure 3.20 that Route 1 is localized from 3rd landmark with known start and from 6th landmark with unknown start. Route 7 is localized given 9 landmarks with unknown start and 3 landmarks with known start. Proposed method is not able to localize Route 3 giving unknown start but localizes route from 2nd landmark with

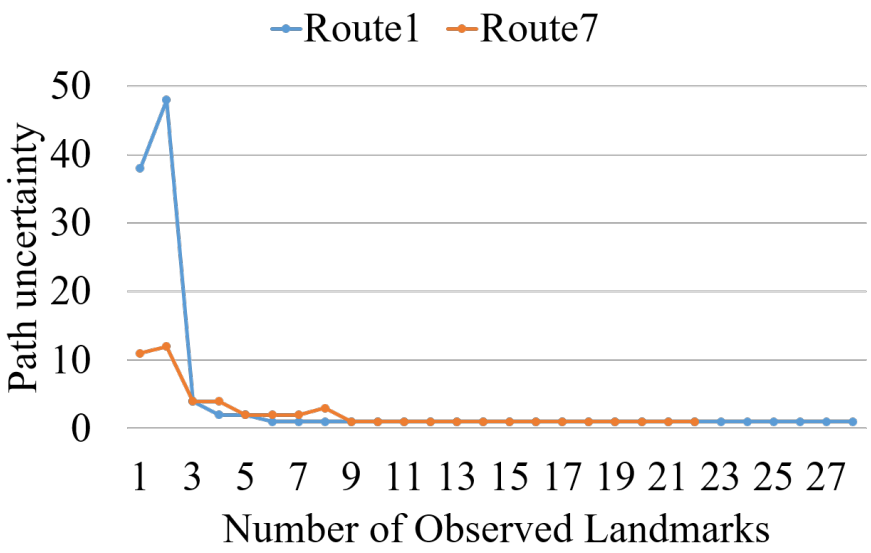
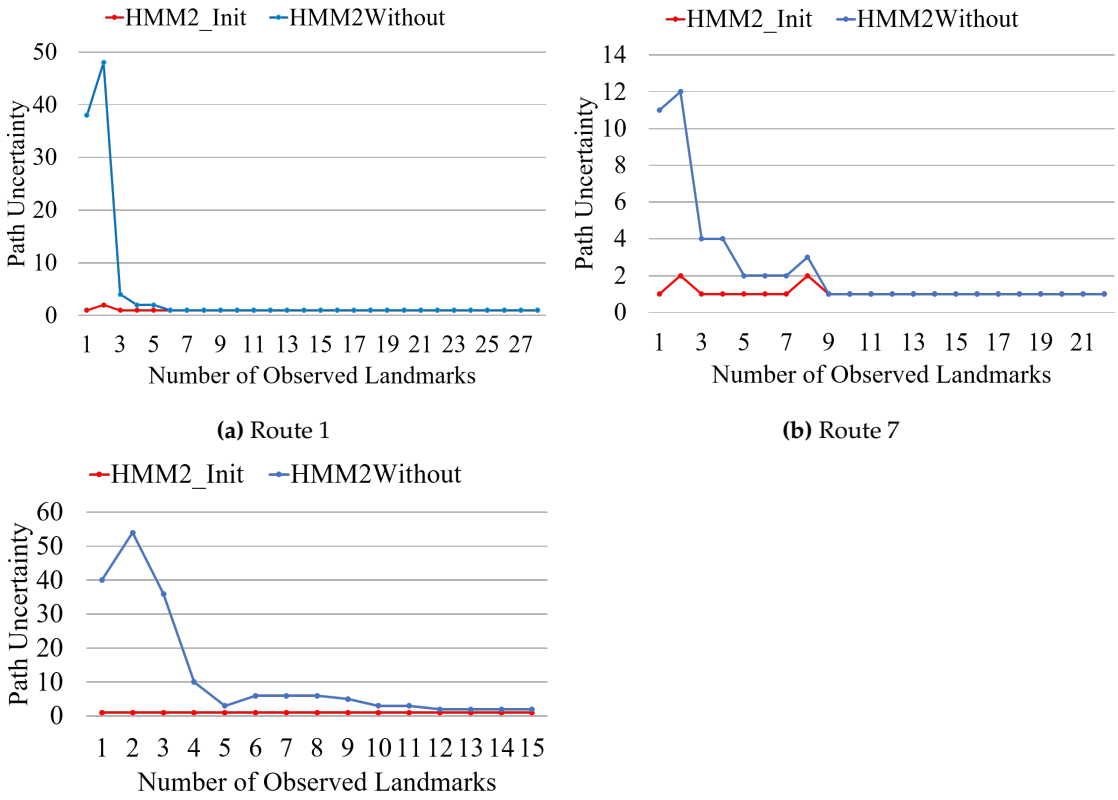


Figure 3.19: Localization performance with the number of observed landmarks in two scenes.

knowing start. It demonstrates that knowing start significantly improves the localization performance in two scenes.



(a) Route 1

(b) Route 7

(c) Route 3

Figure 3.20: Influence of known start on localization results.

3.6 Concluding Remarks

In this thesis, we present a visual landmark sequence-based indoor topological localization method. We propose an innovative representation of landmarks on topological map, a robust landmark detector and an effective sequence matching algorithm for localization. Semantic information of stable indoor elements is exploited to represent the environmental locations. Compared to traditional landmark represented by local key point features, combined geometric elements or text information, our representation is able to stay robust facing dynamic environmental change caused by view changes and image blurring. This high-level representation reduces the storage requirement and can be extended to large indoor environment. We present a robust CNN-based landmark detector for landmark detection. Previous landmark detecting methods are devised based on the predefined rules or colour and gradient information. Slight environment change could significantly influence the landmark detection performance. Background also has significant influence to the detection accuracy. We develop the novel landmark detector using deep learning technique. Instead of designing the feature with the landmark prior, it learns a deep feature representation for landmarks. Experimental results demonstrate that previous designed feature is confused with background while our detector are capable of reliably detecting landmarks from the background.

Our matching algorithm achieves good performance to handle indoor scene ambiguity as it involves more contextual information. Taking objects types as landmark representation saves the storage demand but discards the landmark details. This further increases the scene ambiguity. Methods depending on feature matching fails to work with scene ambiguity problem. HMM helps relieve it in certain degree but still does not solve it. The experiments show that our methods provides better result than HMM to the problem.

For future work, we plan to investigate the fusion of low-level visual features with semantic features, as well as the geometric features. This would decrease the scene ambiguity and require fewer landmarks for localization. Another direction to pursue is to construct the topological map automatically. Currently, we build our topological map manually based on the floor plan map. When there are no floor plan maps of the scenes, constructing the map from massive videos is necessary. Localization approach is not able to handle the situation that the camera stops working for a while as we rely on the landmark occurrence sequence to perform localization. If the camera stops working for a period of time, there will be two video segments. The approach will treat the two video segments as independent videos to perform localization. Two landmark sequences are not able to constrain each other because any number of the landmarks

and any type of the landmarks could be observed during the breaking time.

CHAPTER 4

A Relative Geometry-aware Siamese Neural Network for Image-based Metric Localization

Metric localization, also referred to the 6DOF camera relocalization, is an important component of autonomous driving and navigation. Deep learning has recently emerged as a promising technique to tackle this problem. In this chapter, we present a novel relative geometry-aware Siamese neural network to enhance the performance of deep learning-based methods through explicitly exploiting the relative geometry constraints between images. We perform multi-task learning and predict the absolute and relative poses simultaneously. We regularize the shared-weight twin networks in both the pose and feature domains to ensure that the estimated poses are globally as well as locally correct. We employ metric learning and design a novel adaptive metric distance loss to learn a feature that is capable of distinguishing poses of visually similar images from different locations. We evaluate the proposed method on public indoor and outdoor benchmarks and the experimental results demonstrate that our method can significantly improve localization performance. Furthermore, extensive ablation evaluations are conducted to demonstrate the effectiveness of different terms of the loss function.

The chapter is organized as follows: Section 4.1 describes the camera relocalization. Section 4.2 elaborates the basic idea of deep learning-based camera relocalization methods. Section 4.3 describes the architecture of the proposed network and its loss function items. We present the details of our experiments and evaluation in Section 4.4. Finally, we conclude our work in Section 4.5.

4.1 Introduction

Camera relocalization, or 6 degrees of freedom (6DOF) estimation, refers to the problem of estimating the pose (position and orientation) of an image (camera). It is a hot research topic in structure from motion (SfM), simultaneous localization and mapping (SLAM) and robotics, and it is also an essential component of autonomous driving and navigation.

Global Positioning System (GPS) has been widely used for vehicle localization but its accuracy significantly decreases in urban areas where tall buildings block or weaken its signals. Many image-based methods have been proposed to complement GPS. They provide position and orientation information based either on image retrieval [113, 118, 198–200] or 3D model reconstruction [201]. However, these methods face many challenges, including high storage overheads, low computational efficiency and image variations, especially for large scenes.

Recently, rapid progress in machine learning, particularly deep learning, has produced a number of deep learning-based methods [131–139]. They have attained good performances in addressing the aforementioned challenges but their accuracies are not as good as traditional methods. Another severe problem of deep learning-based methods is that they fail to distinguish two different locations that have similar objects or scenes.

In this thesis, we present a novel relative geometry-aware Siamese neural network, which explicitly exploits the relative geometry constraints between images to regularize the network. We improve the localization accuracy and enhance the ability of the network to distinguish locations with similar images. It is achieved with three key new ideas:

1. We design a novel Siamese neural network that explicitly learns the global poses of a pair of images. We constrain the estimated global poses with the actual relative pose between the pair of images.

2. We perform multi-task learning to estimate the absolute and relative poses simultaneously to ensure that the predicted poses are correct both globally and locally.

3. We employ metric learning and design an adaptive metric distance loss to learn feature representations that are capable of distinguishing the poses of similar visual images of different locations thus improving the overall pose estimation accuracy.

4.2 Deep Learning-based Camera Relocalization

Deep learning-based camera relocalization methods use an end-to-end learning strategy to predict the positions and orientations directly. They do not perform image matching or solve 2D-3D correspondence as traditional methods do. Instead, they regard the task as a regression problem and utilize convolutional neural networks to model the hidden mapping function between the images and their corresponding poses. The networks are supervised by the distance between the predicted poses and the ground truth. This section focuses on discussing the pose representation and describing the loss function formulation.

4.2.1 Pose Representation

The image (camera) pose is comprised of the positional component and the orientational component. The position is denoted by a 3-dimensional vector $\mathbf{x}$ of the arbitrary coordinate space. Orientation can be represented in 3 forms: Euler angle, transformation matrix, and quaternion. Euler angle is not a good choice because it suffers from the gimbals lock problem. Transformation matrix is over-parametrized for orientation because it contains 9 parameters to represent the orientation of 3D space, while the orientation only has 3 degrees of freedom. Previous works [132, 135–137] choose the quaternion to represent orientation, because it is a smooth and continuous representation. The quaternion is a 4-dimensional unit vector $\mathbf{q}$ and is easy to perform back-propagation. The main concern for the quaternion is that each orientation has two different quaternion representations. This can be addressed by constraining the quaternion to one hemisphere.

One simple and obvious way to represent pose is to form a 7-dimensional vector, combining position and orientation together. However, previous works demonstrate that the 7-dimensional vector representation does not achieve good performance due to the difference of scale between position and orientation. Therefore, two pose components are usually regressed separately. In this thesis, instead of training two separate convolutional neural networks to estimate position and orientation, we train one model to predict the two components simultaneously. This is reasonable because both position and orientation come from the same image content.

4.2.2 Loss Function

The loss function (GlobalLoss) is normally designed based on the distance between the predicted pose and the ground truth, serving as the optimization objective for training the networks. It consists of two components, i.e. position loss and orientation loss, as shown in equation (4.2.1).

$$L_G = L_{Gx} + L_{Gq}, \tag{4.2.1}$$

where L_{Gx} is the position loss and L_{Gq} denotes the orientation loss. Here, Euclidean distance is chosen to calculate the position loss and orientation loss as it is continuous and smooth. The two components are computed by equations (4.2.2) and (4.2.3) respectively.

$$L_{Gx} = \|x - \hat{x}\|_2, \tag{4.2.2}$$

where x represents the real position and $\hat{x}$ denotes the predicted one.

$$L_{Gq} = \left\| q - \frac{\hat{q}}{\|\hat{q}\|} \right\|_2, \tag{4.2.3}$$

where q is the ground truth orientation, $\hat{q}$ denotes the predicted orientation and $||\hat{q}||$ represents the length of the predicted orientation quaternion. $\frac{\hat{q}}{\|\hat{q}\|}$ is performed to normalize the predicted quaternion to the length of 1 since the network prediction does not guarantee it.

Due to the quantity and scale difference between the position loss and the orientation loss, a hyperplane parameter β is introduced to balance the influence of the two loss components. The loss function is represented as equation (4.2.4).

$$L = L_{Gx} + \beta \times L_{Gq}, \tag{4.2.4}$$

Previous works choose to set β manually and achieve good performance in their experiments. However, fine tuning β for different scenes is labour-intensive. PoseNet2 addresses this issue by introducing two learnable variables, i.e. $\hat{s}_x$ and $\hat{s}_q$, which correspond to the loss of position and orientation respectively. Then equation (4.2.4) is transformed into equation (4.2.5):

$$L = L_{Gx} \times exp(-\hat{s}_x) + \hat{s}_x + L_{Gq} \times exp(-\hat{s}_q) + \hat{s}_q. \tag{4.2.5}$$

4.3 Relative Geometry-Aware Siamese Network for Camera Relocalization

Our network is built on Siamese network originally introduced by Bromley and LeCun in [202]. A traditional Siamese neural network architecture consists of twin networks which accepts distinct inputs. The loss function computes a metric between the highest-level feature representation on each side given certain threshold. We utilize this structure to learn a robust feature representation for mapping positions and orientations by introducing relative geometry constraints of the training images. The process is supervised by both global pose and relative pose constraints. The proposed network architecture is illustrated in Figure 4.1. Compared to the conventional Siamese network structure, it has an additional component for relative pose prediction and performs multi-task learning. In the following subsections, we will present the network architecture and the relative geometry losses for the network training in detail.

4.3.1 Network Architecture

Each of the twin networks consists of a modified ResNet50 [203] and a global pose regression unit (GPRU). The modified ResNet50 consists of 5 residual blocks and an average pooling layer. Each residual block has multiple residual bottleneck units that are comprised of three convolutional layers with kernel sizes of 1×1, 3×3, and 1×1 in sequence. Each convolutional layer is followed by rectified linear unit (ReLU) and batch normalization operation. The average pooling layer is used to aggregate the feature information from the previous layers. The GPRU contains 3 fully connected layers. The first fully connected layer has 1024 neurons and the followed two has 3 and 4 neurons respectively for regressing the position and orientation. For the relative pose of the two inputs, we design a relative pose regression unit (RPRU). It has a similar structure as the GPRU. The difference lies in their inputs. While the GPRU takes the output vector of the modified ResNet50 as input, the RPRU takes the concatenation of the two modified ResNet50 output vectors as input. The dropout technique is applied after each fully connected layer to reduce feature redundancy. The parameter of dropout layer is set to be 0.2 empirically.

4.3.2 Relative Geometry Losses

We design three relative geometry losses based on the relative geometry constraints of the training images including the relative pose loss (RelLoss), the relative pose regres-

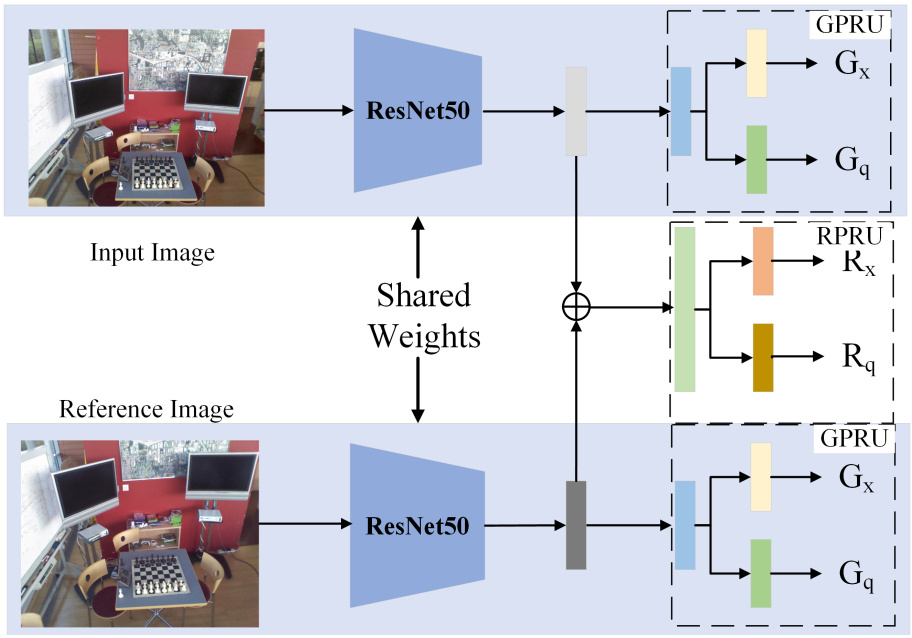


Figure 4.1: Relative Geometry-Aware Siamese neural network architecture for 6DOF camera relocalization. Units of the same colour share the same weights. The silver and grey unit represent the outputs of the modified ResNet50. G_x, G_q denote the positional and orientational components of the predicted global pose, and R_x, R_q denote two components of the predicted relative pose. The global pose regression unit (GPRU) and the relative pose regression unit (RPRU) are represented with dashed-boundary boxes.

sion loss (RelRLoss) and the adaptive metric distance loss (MDLoss). They function in *both* the feature and the pose spaces to regularize the network. They will be discussed in detail in the following sections.

Relative Pose Loss. Previous deep learning-based pose estimation methods train the network on the global poses of the images, i.e. given an input image, they estimate its global (absolute) position and orientation while the relative pose between two training images is ignored. However, the relative pose information of two images is important. In this thesis, the network not only explicitly estimates the global pose of the input image but also explicitly requires that the difference between the estimated global poses of two images is consistent with their actual (ground truth) difference. The relative pose loss (RelLoss) is designed to preserve the relative geometry in the pose space by comparing the distance between two predicted global poses, and the actual distance of the global poses of the two images. RelLoss is able to keep the relative pose of paired images consistent with their ground truth. It works in the pose space and constrains the pose error of two images.

Suppose that the position and orientation of the current image I and a reference image

I_{ref} are (x, q) and (x_{ref}, q_{ref}), respectively. The relative position x_{rel} and orientation q_{rel} can be computed with equations (4.3.1) and (4.3.2).

$$x_{rel} = x - x_{ref}, \tag{4.3.1}$$

$$q_{rel} = q^*_{ref} \times q, \tag{4.3.2}$$

where q^*_{ref} represents the conjugate quaternion of q_{ref}. Note that when calculating the relative orientation from the predicted orientation quaternion with equation (4.3.2), the quaternion has to be normalized. The RelLoss also contains the positional loss component and the orientational loss component as shown in equation (4.3.3).

$$L_C = L_{Cx} + L_{Cq}, \tag{4.3.3}$$

where L_{Cx} denotes the RelLoss positional component, and L_{Cq} is the orientational component.

The two loss components are formulated with Euclidean distance as shown in equations (4.3.4) and (4.3.5).

$$L_{Cx} = \|\hat{x}_{rel} - x_{rel}\|_2, \tag{4.3.4}$$

$$L_{Cq} = \|\hat{q}_{rel} - q_{rel}\|_2, \tag{4.3.5}$$

where $\hat{x}_{rel}, \hat{q}_{rel}$ are the predicted relative position and orientation, and x_{rel}, q_{rel} denote the ground truth.

Relative Pose Regression Loss Whilst RelLoss captures the relative geometry of two images through estimating their global poses, we here introduce another loss to estimate the relative pose distance of a pair of images directly from the input images. The relative pose regression loss (RelRLoss) is defined as shown in equation (4.3.6).

$$L_R = L_{Rx} + L_{Rq}, \tag{4.3.6}$$

where L_{Rx} denotes the positional component, and L_{Rq} denotes the orientational component. The two component loss functions are computed by equations (4.3.7) and (4.3.8).

$$L_{Rx} = \|x_{rel} - \widetilde{x}_{rel}\|_2, \tag{4.3.7}$$

$$L_{Rq} = \left\| q_{rel} - \frac{\widetilde{q}_{rel}}{\|\widetilde{q}_{rel}\|} \right\|_2, \tag{4.3.8}$$

where x_{rel}, q_{rel} represent the ground truth relative position and orientation, and $\widetilde{x}_{rel}, \widetilde{q}_{rel}$ represent the directly predicted relative position and orientation. The ground truth relative position and orientation can be obtained using equation (4.3.1) , (4.3.2). Note that $\widetilde{q}_{rel}$ needs to be normalized as it is directly regressed by the network.

It should be noted that L_R in equation (4.3.6) and L_C in equation (4.3.3) are different. One is computed from the difference of two predicted global poses while the other is predicted directly by regression. Furthermore, it is the L_R that joins the twin networks together (please refer to Figure 4.1). The purposes of introducing RelRLoss is to ensure that the features extracted by the ResNet50 network will not only enable an accurate estimate of the global pose but also an accurate relative pose estimation.

Adaptive Metric Distance Loss. Deep learning-based methods often fail to accurately predict the poses of similar images of different locations. Distinguishing similar inputs belonging to different classes is one of the major difficulties in computer vision. Here, we take advantage of the Siamese network architecture of Figure 4.1 and propose the adaptive metric distance loss (MDLoss) to address the problem. It is inspired by metric learning [204–206]. The basic idea of metric learning is to learn a metric distance adaptive to the problem of interest. For many problems, including camera relocalization, hand-crafted representations fail badly in capturing the notion of similarity. Deep learning regression-based camera relocalization approaches are based on the visual contents of the input image to estimate its pose, therefore simple metrics measuring the visual content similarity fails to capture the pose dissimilarity in the above cases. In the case of our Siamese architecture in Figure 4.1, the 6DOF camera pose is estimated by the GPRU. The input to the GPRU unit (the output of the ResNet50) should reflect the *pose difference* rather than the *visual similarity* of the images. We therefore introduce the adaptive metric distance loss (MDLoss) to address this issue.

The MDLoss is built on the contrastive loss, which employs semantic information (data label) to force the convolutional neural network to learn an embedding representation that complies with a notion of similarity of the problem domain. In our scenario, we define the metric distance loss by embedding the relative pose of two images. The relative information is used to define the margin of feature representation. The loss function is shown in equation (4.3.9).

$$L_{MD} = \frac{1}{2N} \sum_{n=1}^{N} \{\max(d_x + \alpha \times d_q - d, 0)\}^2, \tag{4.3.9}$$

where N denotes the number of the training samples, $d = ||f - f_{ref}||_2$, f and f_{ref} are the outputs of the modified ResNet50 network taken for the current image and the reference image respectively, $d_x = ||x - x_{ref}||_2$ is the Euclidean distance of the actual

relative position while $d_q = ||q - q_{ref}||_2$ is the Euclidean distance of the actual relative orientation of the current image and the reference image, α is a positive constant to balance the influence of the relative position and orientation. It is set equal to 10 empirically.

An explanation of L_{MD} is that, if d is smaller than $d_x + \alpha \times d_q$, we want to make it as large as $d_x + \alpha \times d_q$. On the other hand, if d is larger than $d_x + \alpha \times d_q$, this cost function is not utilized and other cost functions will function to ensure f and f_{ref} to take the appropriate values. This is a reasonable strategy because the reference image is always taken at a different location from that of the current image.

4.3.3 Comprehensive Loss

We train the proposed neural network jointly with GlobalLoss, RelLoss, RelRLoss and MDLoss. The comprehensive loss can be represented by equation (4.3.10).

$$L = L_G + L_C + L_R + L_{MD} \tag{4.3.10}$$

Equation (4.3.10) can also be written in the form of equation (4.3.11).

$$L = L_x + L_q + L_{MD}, \tag{4.3.11}$$

It consists of three components: position loss L_x, orientation loss L_q and metric distance loss L_{MD}. Positional loss and orientational loss each has three components and can be written as equations (4.3.12) and (4.3.13) respectively.

$$L_x = L_{Gx} + L_{Cx} + L_{Rx}, \tag{4.3.12}$$

$$L_q = L_{Gq} + L_{Cq} + L_{Rq}. \tag{4.3.13}$$

We choose a learning strategy to balance the position loss L_x and orientation loss L_q similar to PoseNet2. Therefore, the comprehensive loss can be further reformulated as equation (4.3.14):

$$L = L_x \times \exp(-\hat{s}_x) + \hat{s}_x + L_q \times \exp(-\hat{s}_q) + \hat{s}_q + L_{MD}. \tag{4.3.14}$$

where $\hat{s}_x$ and $\hat{s}_q$ are learnable coefficients.

4.4 Experiments

In this section, we test our method on two publicly available camera relocalization benchmark datasets, one indoor and one outdoor, to demonstrate its effectiveness. Experimental results are presented and compared with state-of-the-art methods in the literatures. We also investigate the role of various components of the loss function and analyze how the choice of reference image affects the performance of the proposed method.

4.4.1 Datasets

The two public datasets we used are: *7Scene* [128] and *Cambridge Landmarks* [132]. To make our results exactly comparable to previous methods, we use the same split of training set and testing set as in the original datasets. The details of the two dataset can be seen in Table 4.1.

7Scene is an indoor image dataset for camera relocalization and trajectory tracking. It is collected with a handhold RGB-D camera. The ground truth pose is generated using the Kinect Fusion approach [207]. The dataset is captured in seven indoor scenes. For each scene, it contains several image sequences, which has already been divided into training and testing sets. The images are taken at the resolution of 640×480 pixel with known focal length of 585. The dataset is quite challenging as motion makes the images blur. Besides, the indoor scenes are usually texture-less, which makes the localization problem even more difficult.

Cambridge Landmarks is an outdoor dataset collected in four sites around Cambridge University. It is collected using a Google mobile phone while pedestrians walk. The images are captured at the resolution of 1920×1080 pixels and the ground truth pose is obtained through VisualSFM software [208]. The dataset is also very challenging as it is taken in different weather and lighting conditions. Besides, the occlusion of moving pedestrians and vehicles further increases the difficulty.

4.4.2 Setup

Training phase: in this phase, all parts of the proposed network are involved. It takes in a pair of images and outputs the corresponding global poses of them. It is important to note that, the twin networks are *identical*. One takes the current image as input and produces its global 6DOF pose information, while the other takes the reference image as input and outputs its corresponding pose.

Table 4.1: The details of the 7Scenes and Cambridge landmark dataset.

Scene	Training	Testing	Spatial scope(m)
Chess	4000	2000	3×2
Fire	2000	2000	2.5×1
Heads	1000	1000	2×0.5
Office	6000	4000	2.5×2
Pumpkin	4000	2000	2.5×2
Red Kitchen	7000	5000	$4 \times$
Stairs	2000	1000	2.5×2
KingsCollege	1220	343	140×40
OldHospital	895	182	50×40
ShopFacade	231	103	35×25
StMarysChurch	1487	530	80×60

Testing phase: in the testing phase, only *one* of the twins is necessary. Since they are identical, any one can be used. The middle part that linking the twins is no longer necessary in this stage. Once training is completed, an image is fed to one of the twin networks and the 6 degree global pose information of the camera can be estimated.

We use the same image pre-processing approaches as previous methods [132]. We firstly resize the image to 256 pixels along the shorter side and normalize it with the mean and standard deviation computed from the ImageNet dataset. For the training phase, we randomly crop the image to 224×224 pixels. For the testing phase, images are cropped to 224×224 pixels at the center of the image. Training images are shuffled before they are fed to the network.

The modified ResNet50 is initialized with pre-trained weights of ImageNet dataset. The GPRU component and the RPRU are initialized with the Xavier initialization [209]. We choose the Adam optimizer to train the network with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The weight decay is 10^{-5} . We train the network with different learning rates from 10^{-3} to 10^{-7} and find that 10^{-5} gives best performance as well as the efficiency. The batch-size is set to be 32 for computational resource reasons. We also initialize the $\hat{s}_x$ and $\hat{s}_q$ with 0 and -3.0 respectively in our experiments. We implement the network with PyTorch and train the network on an Ubuntu 16.04 TS system with a NVIDIA GTX 1080Ti GPU. Training is stopped until the network is converged.

4.4.3 Results

We compare the results of the proposed method with that of state-of-the-art deep learning-based methods such as PoseNet, Bayesian PoseNet, PoseNet2, Hourgrlass-net, LSTM-Net,Vidloc and RelNet on the *7Scene* dataset, and with PoseNet, Bayesian PoseNet, PoseNet2 and LSTM-Net on the *Cambridge Landmarks* dataset. It is perhaps worth mentioning that unlike these methods, VLocNet [141] and VLocNet++ [142] can only work on image sequence and crucially, these two methods require to know the exact pose of the starting frame of the sequence before they can predict the poses of subsequent frames. Therefore, these two methods are not directly comparable with our method and those methods we compare against in this thesis. Similar to others, we report each scene's median error. We also compare the average median accuracy over all scenes in each dataset. The comparative results are shown in Table 4.2 and Table 4.3. Table 4.2

Table 4.2: Comparison of median errors with other deep learning-based methods on the *7Scene* dataset. The reported values are referred to their respective papers.

Scene	PoseNet [132]	Bayesian PoseNet [134]	LSTM-Net [136]	Vidloc [137]	HourglassNet [135]	PoseNet2 [138]	RelNet [140]	Ours (*Median*)	Ours (*Best*)
Chess	0.32m, 8.12°	0.37m, 7.24°	0.24m, 5.77°	0.18m, N/A	0.15m, 6.53°	0.13m, **4.48**°	0.13m, 6.46°	**0.099**m, 5.19°	0.001m, 0.20°
Fire	0.47m, 14.4°	0.43m, 13.7°	0.34m, 11.9°	0.26m, N/A	0.27m, **10.84**°	0.27m, 11.3°	0.26m, 12.72°	**0.253**m, 11.64°	0.001m, 0.15°
Heads	0.29m, 12.0°	0.31m, 12.0°	0.21m, 13.7°	0.14m, N/A	0.19m, **11.63**°	0.17m, 13.0°	0.14m, 12.34°	**0.126**m, 13.20°	0.002m, 0.14°
Office	0.48m, 7.68°	0.48m, 8.04°	0.30m, 8.08°	0.26m, N/A	0.21m, 8.48°	0.19m, **5.55**°	0.21m, 7.35°	**0.161**m, 7.71°	0.001m, 0.10°
Pumpkin	0.47m, 8.42°	0.61m, 7.08°	0.33m, 7.00°	0.36m, N/A	0.25m, 7.01°	0.26m, **4.75**°	0.24m, 6.35°	**0.163**m, 6.61°	0.001m, 0.15°
Redkitchen	0.59m, 8.64°	0.58m, 7.54°	0.37m, 8.83°	0.31m, N/A	0.27m, 10.15°	0.23m, **5.35**°	0.24m, 8.03°	**0.174**m, 8.24°	0.001m, 0.14°
Stairs	0.47m, 13.8°	0.48m, 13.1°	0.40m, 13.7°	0.26m, N/A	0.29m, 12.46°	0.35m, 12.4°	0.27m, **11.82**°	**0.26**m, 13.13°	0.005m, 0.23°
Average	0.44m, 10.4°	0.47m, 9.81°	0.31m, 9.85°	0.25m, N/A	0.23m, 9.53°	0.23m, **8.12**°	0.21m, 9.30°	**0.177**m, 9.39°	0.002m, 0.16°

shows the results for the *7Scene* dataset. It is seen that compared with 7 state-of-the-art deep learning-based camera relocalization methods, the proposed method achieves the best performance on positional accuracy in all 7 scenes. Our method improves the average median positional accuracy by 16% over the best reported result. It is interesting to note that our method has obtained even better result than PoseNet2, which utilizes 3D reference as additional constraints.

For orientational accuracy, we achieve the best result compared to methods based on direct regression. It is not surprising that the results are not as good as PoseNet2 and RelNet since PoseNet2 requires additional 3D models and RelNet triangulates the pose with all referencing images by estimating the relative poses instead of directly regressing results.

Table 4.3 shows the results for the *Cambridge Landmarks* dataset. It can be seen that our method obtains the best positional accuracy on the KingsCollege and the ShopFacade scenes, reaching accuracies of 0.865m and 0.834m respectively. We improve the state-of-the-art orientational accuracy of the OldHospital and the StMarysChurch scenes from 3.29° and 3.32° to 2.42° and 2.98°, achieving 26% and 10% improvement

Table 4.3: Comparison of median errors with other deep learning-based methods on the *Cambridge Landmarks* dataset. The reported values are referred to their respective papers.

Scene	PoseNet [132]	Bayesian PoseNet [134]	LSTM-Net [136]	PoseNet2 [138]	Ours (*Median*)	Ours (*Best*)
KingsCollege	1.92m, 5.40°	1.74m, 4.06°	0.99m, 3.68°	0.88m, **1.04**°	**0.865**m, 1.96°	0.036m, 0.01°
OldHospital	2.31m, 5.38°	2.57m, 5.14°	**1.51**m, 4.29°	3.20m, 3.29°	1.617m, **2.42**°	0.113m, 0.01°
ShopFacade	1.46m, 8.08°	1.25m, 7.54°	1.18m, 7.44°	0.88m, **3.78**°	**0.834**m, 5.56°	0.045m, 0.01°
StMarysChurch	2.65m, 8.46°	2.11m, 8.38°	**1.52**m, 6.68°	1.57m, 3.32°	1.650m, **2.98**°	0.087m, 0.01°
Average	2.08m, 6.83°	1.92m, 6.28°	1.30m, 5.52°	1.62m, **2.86**°	**1.24**m, 3.23°	0.070m, 0.01°

respectively. The average positional accuracy over all scenes is improved from 1.30m to 1.24m. The average orientational accuracy over all scenes is only a little worse than that of PoseNet2, which is trained with 3D model constraints.

It is interesting to note that of all the methods presented in the two tables, some did better in positional accuracy and some did better in orientational accuracy, none of them seems to comprehensively beat the others in both measures. Our method achieves the best average positional accuracy amongst all methods in both datasets. For orientational accuracy, our method achieves competent results, which is only slightly worse than the best method (PoseNet2) but better or at least as good as the other methods.

4.4.4 Discussion

In this section, we perform analysis on the influence of various loss function components and the reference image selection strategy. The experiments are also done on the *7Scene* and *Cambridge Landmarks*.

Loss Analysis. We perform ablation analysis on the loss function. Recall from equation (4.3.10), the overall loss function is $L = L_G + L_C + L_R + L_{MD}$, consisting of the the global loss L_G, the relative pose loss L_C, the relative pose regression loss L_R, and the adaptive metric distance loss L_{MD}. In order to assess the role these loss components play, we formulate 4 loss functions based on the following combinations:

1. G: GlobalLoss;
2. G+C: GlobalLoss + RelLoss;
3. G+C+R: GlobalLoss + RelLoss + RelRLoss;
4. Ours: GlobalLoss + RelLoss + RelRLoss + MDLoss.

We train the proposed network by the 4 aforementioned loss functions separately. The results are shown in Table 4.4 for the *7Scene* dataset and in Table 4.5 for *Cambridge landmarks*. It is seen that as more loss terms are added to the loss function, both positional

Table 4.4: Comparison of different loss combinations with median error on *7Scene* dataset.

Scene	G	G+C	G+C+R	Ours
Chess	0.135m, 7.62°	0.118m, **5.10**°	0.116m, 6.50°	**0.099**m, 5.19°
Fire	0.285m, 13.13°	0.258m, 12.93°	0.258m, 12.48°	**0.253**m, **11.64**°
Heads	0.185m, 14.01°	0.140m, 14.77°	0.144m, 13.82°	**0.126**m, **13.20**°
Office	0.180m, 8.18°	0.173m, **7.65**°	0.175m, 8.19°	**0.161**m, 7.71°
Pumpkin	0.215m, 7.77°	0.226m, 7.87°	0.214m, 6.80°	**0.163**m, **6.61**°
Redkitchen	0.266m, **8.21**°	0.253m, 9.20°	0.201m, 8.24°	**0.174**m, 8.24°
Stairs	0.345m, 13.51°	0.324m, **12.07**°	0.279m, 13.18°	**0.260**m, 13.13°
Average	0.230m, 10.34°	0.213m, 9.94°	0.198m, 9.89°	**0.177**m, **9.39**°

error and orientational error decrease for all scenes of the *7Scene* dataset and the *Cambridge Landmarks* dataset. The average positional error and orientational error for the *7Scene* dataset and *Cambridge Landmarks* dataset are shown in Figure 4.2 and in Figure 4.3 respectively. We can see that average position and orientation errors show a decreasing trend by adding more constraints. This demonstrates the usefulness of each loss component combinations.

Table 4.5: Comparison of different loss combinations with median error on *Cambridge Landmarks* dataset.

Scene	G	G+C	G+C+R	Ours
KingsCollege	1.07m, 4.22°	0.932m, 2.69°	0.97m, 2.14°	**0.865**m, **1.96**°
OldHospital	1.76m, 4.97°	1.650m, 3.38°	1.67m, 3.01°	**1.617**m, **2.42**°
ShopFacade	1.00m, 6.65°	0.930m, 6.23°	0.858m, 5.92°	**0.834**m, **5.56**°
StMarysChurch	1.76m, 4.03°	1.720m, 4.06°	1.684m, 4.83°	**1.615**m, **2.98**°
Average	1.396m, 4.97°	1.308m, 4.09°	1.296m, 3.98°	**1.242**m, **3.23**°

Comparison of Relative Geometry Losses. We have designed three relative geometry-based losses. In order to evaluate their performance separately for pose prediction, we formulate new losses by combining each of them with the global pose loss. We also use global pose loss and our comprehensive loss as baselines. The details of the loss combinations are listed as follows:

1. G : GlobalLoss;
2. G+M: GlobalLoss + MDLoss;
3. G+C: GlobalLoss + RelLoss;

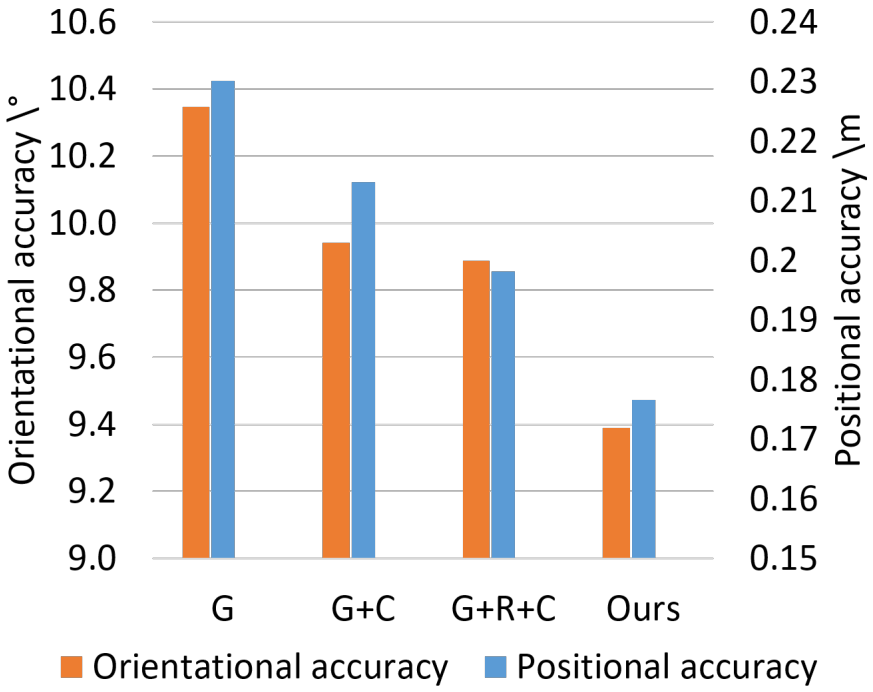


Figure 4.2: The loss analysis over the average errors on *7Scene* dataset.

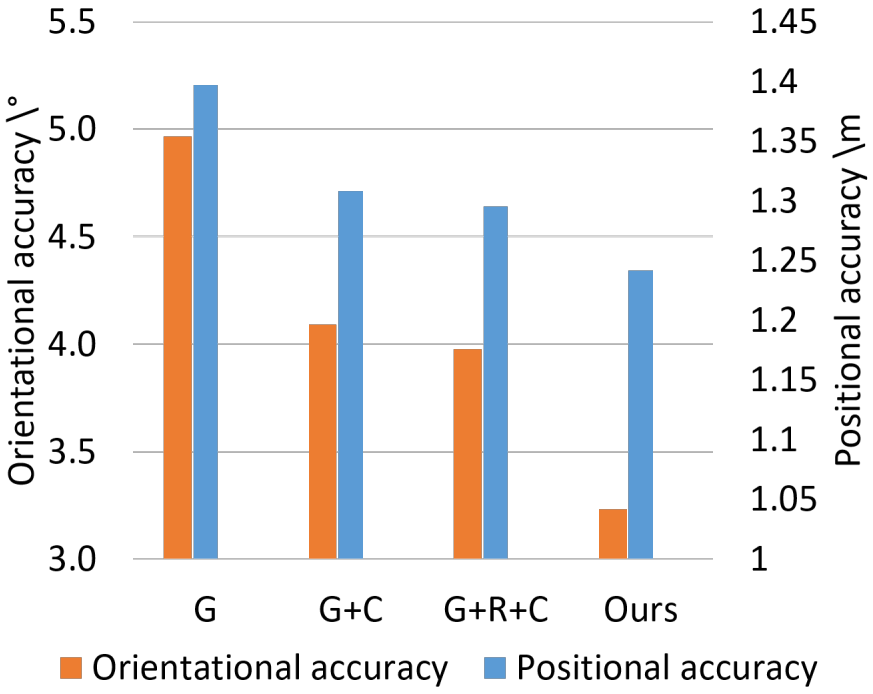


Figure 4.3: The loss analysis over the average errors on *Cambridge Landmarks* dataset.

Table 4.6: Evaluation of each relative loss function with median error on *7Scene* dataset.

Scene	G	G+M	G+C	G+R	Ours
Chess	0.135m, 7.62°	0.116m, **4.82**°	0.118m, 5.10°	0.117m, 5.05°	**0.099**m, 5.19°
Fire	0.285m, 13.13°	0.271m, 11.91°	0.258m, 12.93°	0.262m, 12.64°	**0.253**m, **11.64**°
Heads	0.185m, 14.01°	0.128m, 13.37°	0.140m, 14.77°	0.147m, 13.21°	**0.126**m, **13.20**°
Office	0.180m, 8.18°	0.177m, 7.17°	0.173m, 7.65°	0.189m, **7.13**°	**0.161**m, 7.71°
Pumpkin	0.215m, 7.77°	0.198m, 6.26°	0.226m, 7.87°	0.196m, **5.82**°	**0.163**m, 6.61°
Redkitchen	0.266m, 8.21°	0.217m, **7.55**°	0.253m, 9.20°	0.218m, 7.79°	**0.174**m, 8.24°
Stairs	0.345m, 13.51°	0.265m, 11.98°	0.324m, 12.07°	0.281m, **11.49**°	**0.260**m, 13.13°
Average	0.230m, 10.34°	0.196m, **9.01**°	0.213m, 9.94°	0.201m, 9.02°	**0.177**m, 9.39°

4. G+R: GlobalLoss + RelRLoss;

5. Ours: GlobalLoss + RelLoss + RelRLoss + MDLoss.

For each loss function, we repeat experiments using the same training setup in previous experiments. The results on *7Scene* and on *Cambridge Landmarks* are shown in Table 4.6 and in Table 4.7 respectively. The average localization errors of the two datasets are shown in Figure 4.4 and Figure 4.5.

Table 4.7: Evaluation of each relative loss function with median error on *Cambridge Landmarks* dataset.

Scene	G	G+M	G+C	G+R	Ours
KingsCollege	1.07m, 4.22°	0.960m, 2.79°	0.932m, 2.69°	0.980m, 2.31°	**0.865**m, **1.96**°
OldHospital	1.76m, 4.97°	1.650m, 3.31°	1.650m, 3.38°	**1.615**m, 3.77°	1.617m, **2.42**°
ShopFacade	1.00m, 6.65°	0.876m, **5.11**°	0.930m, 6.23°	0.868m, 5.19°	**0.834**m, 5.56°
StMarysChurch	1.76m, 4.03°	1.617m, 5.83°	1.720m, 4.06°	1.664m, 4.68°	**1.615**m, **2.98**°
Average	1.396m, 4.97°	1.275m, 4.257°	1.308m, 4.09°	1.282m, 4.06°	**1.242**m, **3.23**°

As shown in the two Figures, relative geometry-related losses (G+M, G+C, G+R) achieve better accuracy than global pose alone in most scenes of the two datasets. This further demonstrates their effectiveness on global pose prediction. It can also be seen that G+M obtains a larger average accuracy increase compared with the other two in average. In addition, G+C acquires the smallest accuracy improvement on both datasets, lower than G+R. This implies that relative geometry constraints work better in feature space than in the pose space since RelRLoss and MDLoss are in the feature space while RelLoss is in the pose space. It should also be noted that the results of our proposed loss (G+C+R+M) outperforms all the other single relative geometry-related losses, which further demonstrate the effectiveness of our comprehensive loss function.

Comparison of Metric Losses. To further evaluate the proposed adaptive metric dis-

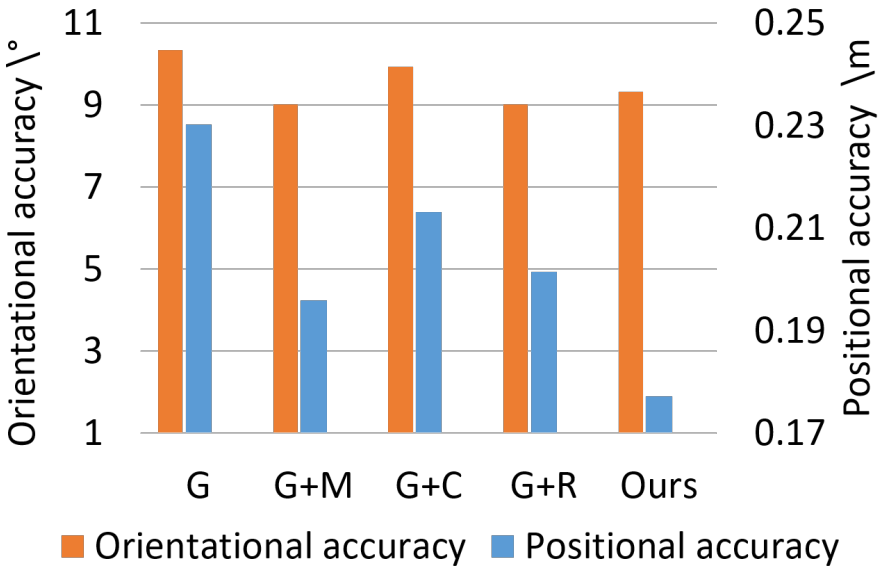


Figure 4.4: The relative loss analysis over the average errors on *7Scene* dataset.

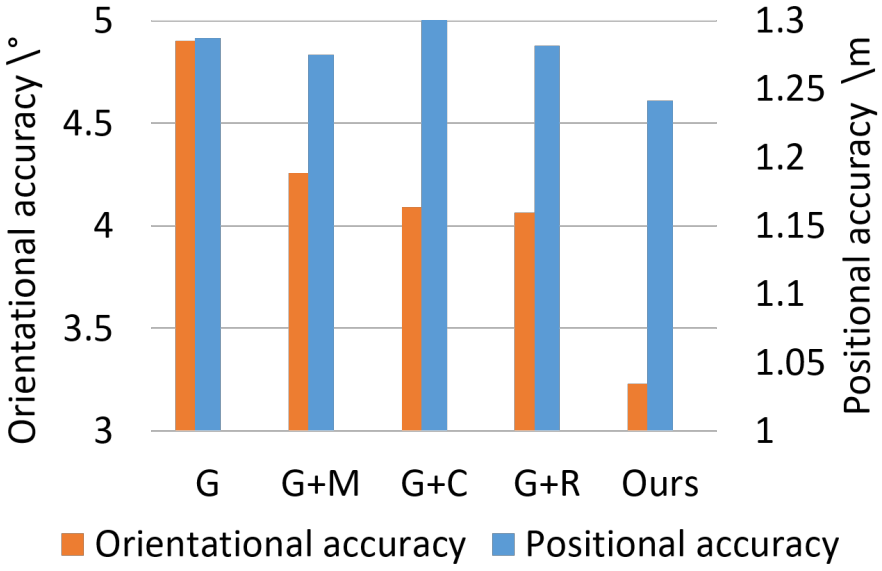


Figure 4.5: The relative loss analysis over the average errors on *Cambridge Landmarks* dataset.

tance loss, we conduct experiments to compare it with conventional Siamese loss [210] and triplet loss [211], since the two losses can also help make visually similar image distinctive as the proposed metric distance loss does. The Siamese loss is shown in equation (4.4.1) and the triplet loss is shown in equation (4.4.2). The major difference is that the conventional metric losses set the margin to be a fixed value while our loss is a function of the relative pose of two images.

$$L_{Siamese} = \frac{1}{2N}\sum_{n=1}^{N}\{(1-y)d^2 + y\{max(m-d,0)\}^2\}, \tag{4.4.1}$$

$$L_{Triplet} = \sum_{n=1}^{N}[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + m]_+ \tag{4.4.2}$$

where $[x]_+$ represents $max(x, 0)$ as hinge loss, N is the number of training samples, m is the margin, d is the feature distance of the paired image, y always equals 1, since the two images are not from the same location. $f(x_i^a)$, $f(x_i^p)$ and $f(x_i^n)$ are the feature vectors of the ith training image, its reference images, and the image after the reference image, respectively. In the Siamese loss of equation (4.4.1), it explicitly forces the features of the two images to be different because they are from two different locations. In the triplet loss (4.4.2), it explicitly enforces that the difference between the ith image and its reference should be smaller than the difference between it and the image after the reference image. In the experiments, we simply replace the MDLoss with the Siamese loss and the triplet loss respectively and repeat the experiment. The margin parameter m of the Siamese loss and the triplet loss is empirically set to be 0.001, which gives the best accuracy. Three comparative losses are listed as below.

1. LossSiamese: GlobalLoss + RelLoss + RelRLoss + SiameseLoss;
2. LossTriplet: GlobalLoss + RelLoss + RelRLoss + TripletLoss;
3. Ours: GlobalLoss + RelLoss + RelRLoss + MDLoss.

We repeat the experiments on the two datasets using the above losses and the results are shown in Table 4.8 for *7Scene* and Table 4.9 for *Cambridge Landmarks*. The average localization errors of the two datasets are shown in Figure 4.6 and Figure 4.7.

It can be seen that our method achieves the best average position accuracy on *7Scene* dataset, and both average position accuracy and average orientation accuracy on the *Cambridge Landmarks* dataset. The LossSiamese acquires the best orientational accuracy on the *7Scene* dataset. LossTriplet performs badly on the *7Scene* dataset but obtains better performance than LossSiamese on the *Cambridge Landmarks* dataset. Although

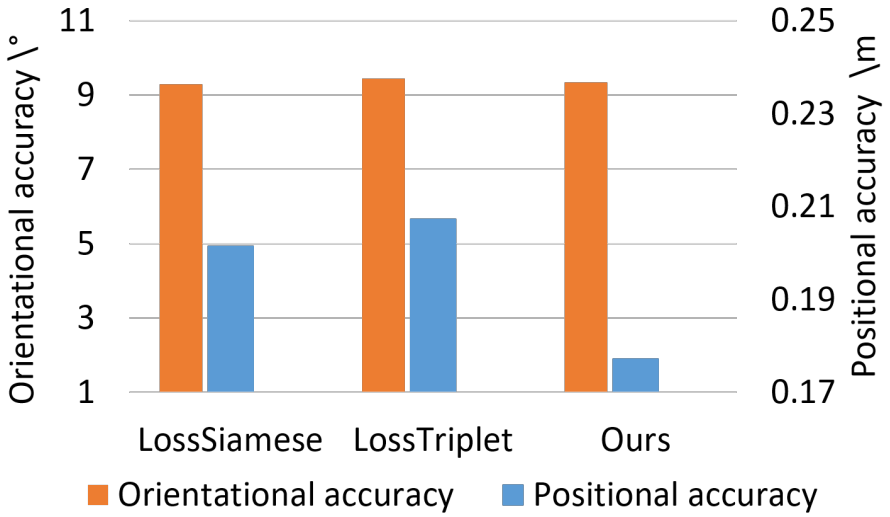


Figure 4.6: The metric loss analysis over the average errors on *7Scene* dataset.

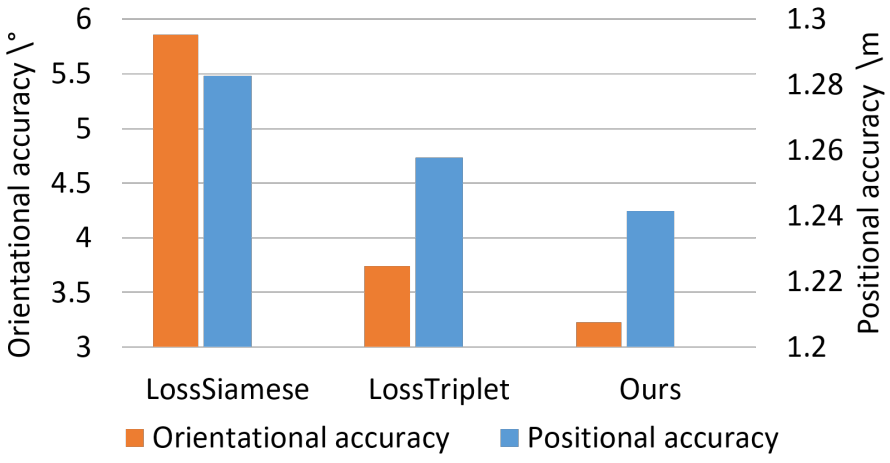


Figure 4.7: The metric loss analysis over the average errors on *Cambridge Landmarks* dataset.

Table 4.8: Comparison between metric loss functions and adaptive metric distance loss with median error on *7Scene* dataset.

Scene	LossSiamese	LossTriplet	Ours
Chess	0.127m, **5.17**°	0.139m, 5.42°	**0.099**m, 5.19°
Fire	0.273m, 13.57°	0.276m, 12.94°	**0.253**m, **11.64**°
Heads	0.128m, 13.45°	**0.125**m, 14.76°	0.126m, **13.20**°
Office	0.188m, 7.77°	0.192m, 7.60°	**0.161**m, **7.71**°
Pumpkin	0.198m, 6.13°	0.216m, **6.03**°	**0.163**m, 6.61°
Redkitchen	0.219m, 8.32°	0.224m, 8.30°	**0.174**m, **8.24**°
Stairs	0.277m, **10.59**°	0.279m, 11.07°	**0.260**m, 13.13°
Average	0.201m, **9.29**°	0.207m, 9.44°	**0.177**m, 9.39°

Table 4.9: Comparison between metric loss functions and adaptive metric distance loss with median error on *Cambridge Landmarks* dataset.

Scene	LossSiamese	LossTriplet	Ours
KingsCollege	0.867m, 4.87°	**0.839**m, 2.03°	0.865m, **1.96**°
OldHospital	1.675m, 5.73°	1.683m, 3.82°	**1.617**m, **2.42**°
ShopFacade	0.861m, 5.76°	0.847m, **5.01**°	**0.834**m, 5.56°
StMarysChurch	1.728m, 7.06°	1.650m, 4.10°	**1.615**m, **2.98**°
Average	1.282m, 5.86°	1.258m, 3.74°	**1.242**m, **3.23**°

the LossSiamese achieves the best orientational accuracy, our method obtains more best performances of the two datasets shown in Table 4.8 and in Table 4.9. The results show that our adaptive metric distance loss outperforms the conventional Siamese loss and the triplet loss.

Reference Image Analysis. In this section, we evaluate two strategies of choosing the reference image. One obvious strategy is to pair every two different images, but it will result in exponential increase of the training time and high information redundancy. To make the training phase efficient, we generate only one reference image for each image. Specifically, reference images are selected in two ways: 1) select the next image in the same image sequence as the reference image; 2) randomly select a different image of the dataset that is not a reference image of any other images. It should be noted that the next image is visually similar to the current image. Randomly chosen reference image has no such property. To evaluate the effectiveness of the two reference image selection strategies on the adaptive metric loss (MDLoss), we train the proposed network with the comprehensive loss function. In addition, we use the result of the networks trained without MDLoss (G+R+C) as baseline to compare the results.

Table 4.10: Comparison of median errors of two reference image selection strategies on the *7Scene* dataset.

Scene	G+R+C	Random	Next
Chess	0.116m, 6.50°	0.109m, 5.46°	**0.099**m, **5.19**°
Fire	0.258m, 12.48°	0.265m, 12.54°	**0.253**m, **11.64**°
Heads	0.144m, 13.82°	0.138m, 13.72°	**0.126**m, **13.20**°
Office	0.175m, 8.19°	0.172m, 8.17°	**0.161**m, **7.71**°
Pumpkin	0.214m, 6.80°	0.207m, **6.33**°	**0.163**m, 6.61°
Redkitchen	0.201m, 8.24°	0.202m, 8.89°	**0.174**m, **8.24**°
Stairs	0.279m, 13.18°	0.287m, **11.89**°	**0.260**m, 13.13°
Average	0.198m, 9.89°	0.197m, 9.57°	**0.177**m, **9.39**°

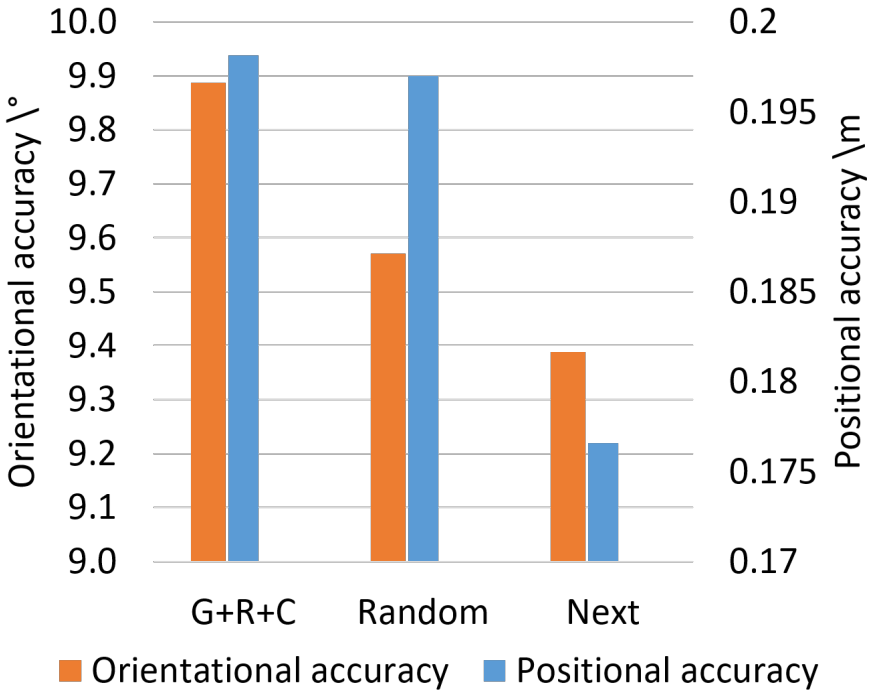


Figure 4.8: The average median errors of two reference image selection strategies on *7Scene* dataset.

Comparative median error results for the different reference selection strategies for the *7Scene* and *Cambridge Landmarks* are shown in Table 4.10 and Table 4.11. The average positional and orientational errors are shown in Figure 4.8 and in Figure 4.9. It can be seen that strategy of choosing the next image as reference image obtains higher image similarity score than that of randomly choosing in two datasets since it achieves lower feature distance.

From Table 4.10, it is seen that compared to the random reference selection strategy, taking the next image as reference image increases the average positional accuracy from 0.197m to 0.177m and the average orientational accuracy from 9.57° to 9.39°. It is also seen that for both reference image selection strategies, the inclusion of MDLoss improves performance. One probable explanation is that MDLoss makes the network learn to keep similar images of different poses apart in the feature space.

Table 4.11: Comparison of median errors of two reference image selection strategies on *Cambridge Landmarks* dataset.

Scene	G+C+R	Random	Next
KingsCollege	0.970m, 2.14°	1.120m, 2.09°	**0.865**m, **1.96**°
OldHospital	1.670m, 3.01°	1.618m, 2.80°	**1.617**m, **2.42**°
ShopFacade	0.858m, 5.92°	1.000m, **4.91**°	**0.834**m, 5.56°
StMarysChurch	1.684m, 4.83°	1.714m, 3.26°	**1.650**m, **2.98**°
Average	1.296m, 3.98°	1.363m, 3.26°	**1.242**m, **3.23**°

As shown in Table 4.11, the results of taking the next image as reference are better than that of the random reference selection strategy on both the average positional and orientational accuracy. It is also seen that randomly choosing the reference image achieves the worse performance on positional accuracy than the baseline. This may be explained by the fact that images of the *Cambridge Landmarks* are of large difference so that the metric distance loss fails to work. To verify the explanation, we measure image similarity of the two pairing strategies. The average Euclidean distance of GIST features of paired images are employed to quantify paired image similarity. The average feature distances of the scenes are shown in Figure 4.10.

Table 4.12 shows that for each scene, taking the next image as reference achieves higher similarity between paired images than that of randomly chosen. This confirms our explanation that MDLoss works better in scenarios where paired images are similar.

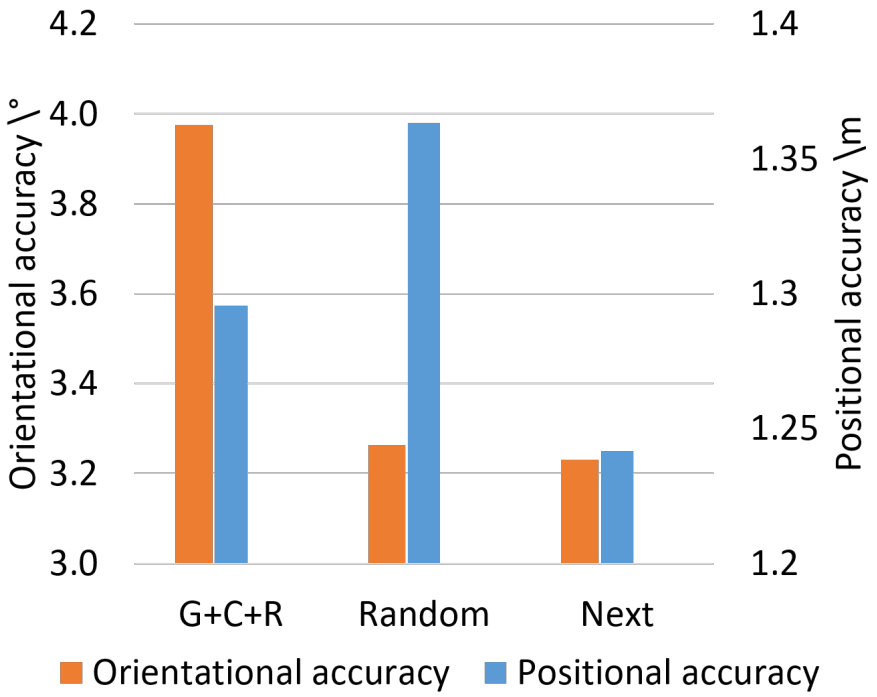


Figure 4.9: The average median errors of two reference image chosen strategies on *Cambridge Landmarks*.

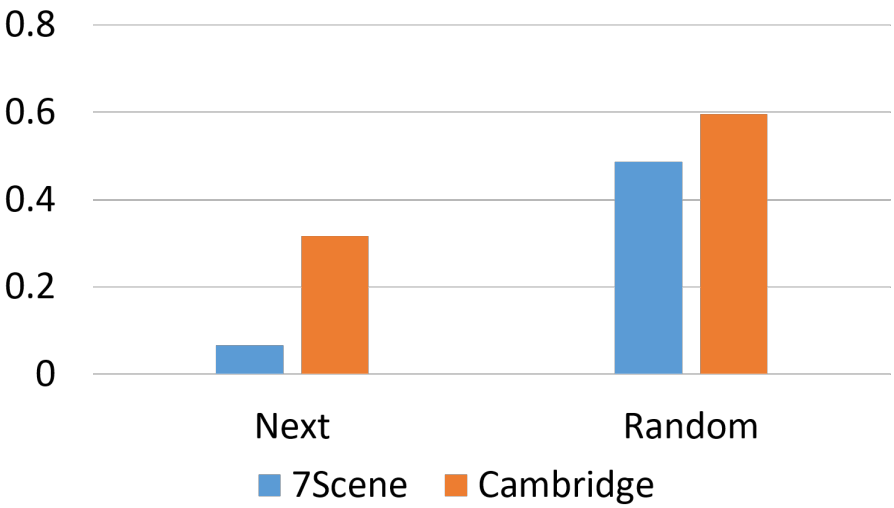


Figure 4.10: Average paired image similarity (measured with average Gist feature distance) on two datasets.

Table 4.12: Statistic of image similarity (measured by average Gist features distance) of two image pairing strategies.

Scene	Training	Testing	Spatial scope(m)	Next	Random
Chess	4000	2000	3×2	0.0700	0.5044
Fire	2000	2000	2.5×1	0.0968	0.5187
Heads	1000	1000	2×0.5	0.0613	0.4866
Office	6000	4000	2.5×2	0.0600	0.4366
Pumpkin	4000	2000	2.5×2	0.0540	0.4390
Red Kitchen	7000	5000	4×3	0.0749	0.4993
Stairs	2000	1000	2.5×2	0.0540	0.5220
KingsCollege	1220	343	140×40	0.2816	0.5531
OldHospital	895	182	50×40	0.3338	0.6127
ShopFacade	231	103	35×25	0.3133	0.5730
StMarysChurch	1487	530	80×60	0.3411	0.6471

4.5 Concluding Remarks

In this thesis, we enhance the camera relocalization performance of deep learning-based methods by introducing the relative geometry constraints. This is achieved by designing a relative geometry-aware Siamese neural network and three relative geometry-related loss functions. The proposed network is capable of predicting the poses of two images as well as the relative pose between them. Another advantage of the network is that it is able to predict the global pose by feeding a single image into one stream of it. The new pose space relative loss and feature space relative regression loss functions can be combined with traditional global pose loss to enhance the position and orientation accuracy. The metric distance loss enables the network to learn deep feature representation that can distinguish similar images of different locations, thus helping improve localization accuracy. We also find that pairing similar images outperforms random paring. Most of time are spent on training the networks for pose regression. The training time varies according to the size of the training images. The test phase barely costs time especially in GPU mode. In future work, we plan to investigate the combination of deep learning-based methods and 3D modelling-based methods to further enhance the performance.

In the following chapter, we will describe our learning-based metric pose regression method through deep convolutional neural networks, which fully takes advantages of the training datasets.

CHAPTER 5

Single Image-based Indoor Metric Localization in 3D Maps

Image localization is an important supplement to GPS-based methods, especially in indoor scenes. Traditional methods depending on image retrieval or structure from motion (SfM) techniques either suffer from low accuracy or fail to work due to the texture-less or repetitive indoor surfaces. With the development of LiDAR technologies, 3D maps are easily constructed in indoor scenes. Image-based indoor localization within a 3D map is a timely but unsolved research problem. In this chapter, we present a new approach to addressing single indoor image localization. In contrast to previous methods that require multiple overlapping images or videos, our new approach can achieve high localization accuracy using only a single image. We achieve this through estimating the depth map of the input image and performing geometry matching in the 3D space. We have developed a novel depth estimation method by utilizing both the 3D map and RGB images where we use the RGB image to estimate a dense depth map and use the 3D map to guide the depth estimation. We will show that our new method significantly outperforms current RGB image based depth estimation methods for both indoor and outdoor datasets. We also show that utilizing the depth map predicted by the new method for single indoor image localization can improve both position and orientation localization accuracy over state-of-the-art methods.

This chapter is organized as follows: Section 5.1 describes the problem of single image localization in a 3D map. Section 5.2 describes each component of the proposed single indoor image localization approach. Section 5.3 elaborates the details of the proposed depth prediction method. Experimental results are shown in Section 5.4. Finally, we conclude our work in Section 5.5.

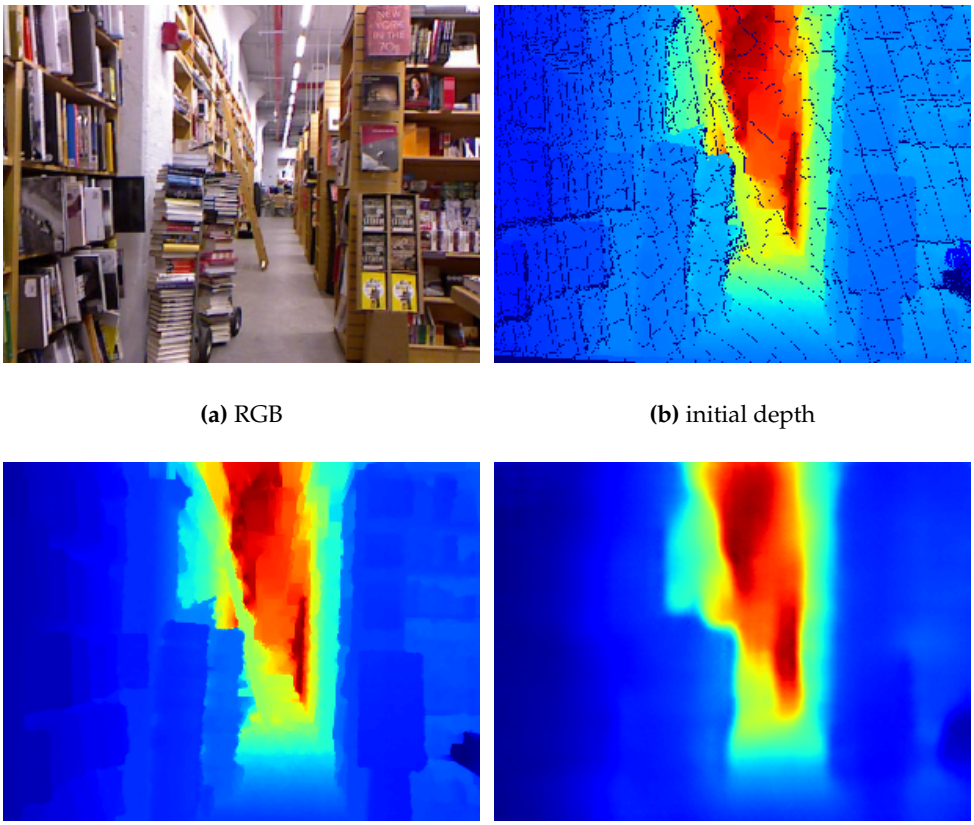
5.1 Introduction

Single image localization is a promising alternative to GPS for indoor localization as GPS signals are mostly blocked in indoor environments. It is also a key component of many computer vision tasks like structure from motion (SfM), simultaneous localization and mapping (SLAM) as well as many applications such as robotics and autonomous driving. It refers to the problem of estimating the 6 DoF parameters of the query image. Traditional methods address it either through image matching [106, 107, 118] or constructing point-to-point associations between the query images and a 3D model built with SfM algorithms [124–126]. However, they are not feasible for many indoor scenes as image matching-based methods are not accurate, and 3D models are difficult to construct using SfM if the environment is comprised of textureless surfaces like white walls or repetitive decoration.

The rapid development of LiDAR instruments makes it easy to build 3D model of indoor scenes as it only relies on the geometry information without any requirements for the surface. However, point-to-point matching methods still do not function on 3D models built from LiDAR sensors which lack the colour information compared to that generated from SfM. Image localization in a LiDAR map is a hot research topic due to widely available 3D LiDAR maps and cameras embedded on the smart phones. Directly matching 2D images and 3D LiDAR model is a very challenging problem as image geometry are ambiguous compared to 3D LiDAR models. Two strategies can be used to tackle it: (1) matching in 2D space; (2) matching in 3D space. Methods based on matching in 2D space are similar to image retrieval methods. The key problem of them is to design similarity metric between two source information. Another strategy is to match in 3D space. It infers depth of RGB images to generate the 3D point cloud and matches against LiDAR map through 3D geometry matching. The key problem of it is to accurately estimate the RGB image depth. Traditional methods estimate the image depth with SfM algorithms and require multiple overlapping images [212, 213]. But they fail in low-texture indoor scenes. Besides, it is time-consuming to estimate the dense depth.

In this thesis, we present a new approach to addressing single image localization in 3D LiDAR maps through depth inference from RGB images. The proposed method exploits the deep learning technique to perform single image depth inference and localizes the query images based on 3D geometry alignment. Our method firstly estimates the coarse pose using our previous work on camera 6DoF relocalization in previous chapter and we warp the 3D map into an initial depth image of the RGB images. Instead of only inferring from RGB image, we predict depth from RGB image with the

initial depth as well. Attaching additional depth information is able to enhance the depth prediction performance as it gives an initial guidance to the RGB images instead of being totally blind. An example is shown in Figure 5.1. Compared to the real depth map, the initial depth is sparser and the structure has tiny misalignment. Given the predicted dense depth, a 3D point cloud is generated and aligned to the 3D LiDAR map to finally localize the image. The process of the proposed method is illustrated in Figure 5.3. The whole approach is a coarse-to-fine process. At first, we estimate a coarse pose with deep regression as proposed in previous chapter [214]. Then, the ICP algorithm is used to align the point cloud produced from predicted depth to correct the initial pose.



(a) RGB **(b)** initial depth

(c) ground truth **(d)** prediction

Figure 5.1: A deep learning-based RGB image depth prediction approach with the guidance of initial depth image generated from a 3D map.

In summary, we make the following contributions in this chapter:

1. We present a new approach for image-based indoor localization in a 3D map. In contrast to previous methods that require multiple overlapping images or videos, our new approach can achieve high localization accuracy using only a single im-

age. We achieve this through estimating the depth map of the input image and performing geometry matching in the 3D space.

2. We propose a novel depth estimation approach by utilizing both the 3D map and RGB images. We use the 3D map to generate an initial depth map and thus guide the RGB image to produce a fine depth map. Our new method significantly outperforms current RGB image based depth estimation methods for both indoor and outdoor datasets.

3. We present extensive experimental results to demonstrate the effectiveness of our new depth estimation method and the new single indoor image localization approach.

The rationale behind our approach is that we believe monocular depth prediction and image-based 3D localization are two interleaved problems: once the depth is accurately predicted, the image localization accuracy should be on par with the 3D point cloud registration approaches; once the image is accurately localized, it should generate an accurate depth prediction result that is well aligned with the 3D point cloud map.

5.2 Single Image Localization within a 3D Map

We consider a scenario as shown in Figure 5.2 where we are given a single 2D RGB indoor colour image and the 3D map of the scene and our aim is to estimate the pose (6 DoF) of the 2D image. While past research has considered the case where multiple overlapping images or a video is available, we consider the more challenging case that only a single RGB image is available. It is a very difficult problem as it tries to register an image to a point cloud, as each comes from a different modality. Images contain colour information in 2D space and the point cloud contains the geometric information. Since no colour information can be utilized from point cloud, we propose an approach to addressing the problem by inferring the geometry information for 2D colour images. It is achieved by predicting the corresponding depth images of the RGB images. Given the depth images, we can obtain point clouds from the them. Then, the iterative closest point (ICP) algorithm is applied to register the produced point clouds with the 3D map through geometry matching.

This section describes the proposed approach of localizing a single image in a 3D LiDAR map. Figure 5.3 shows the process of the proposed method, including four steps: pose initialization, local map extraction, point cloud generation and ICP-based geometry matching. Pose initialization step provides a coarse pose. Given the coarse pose, we

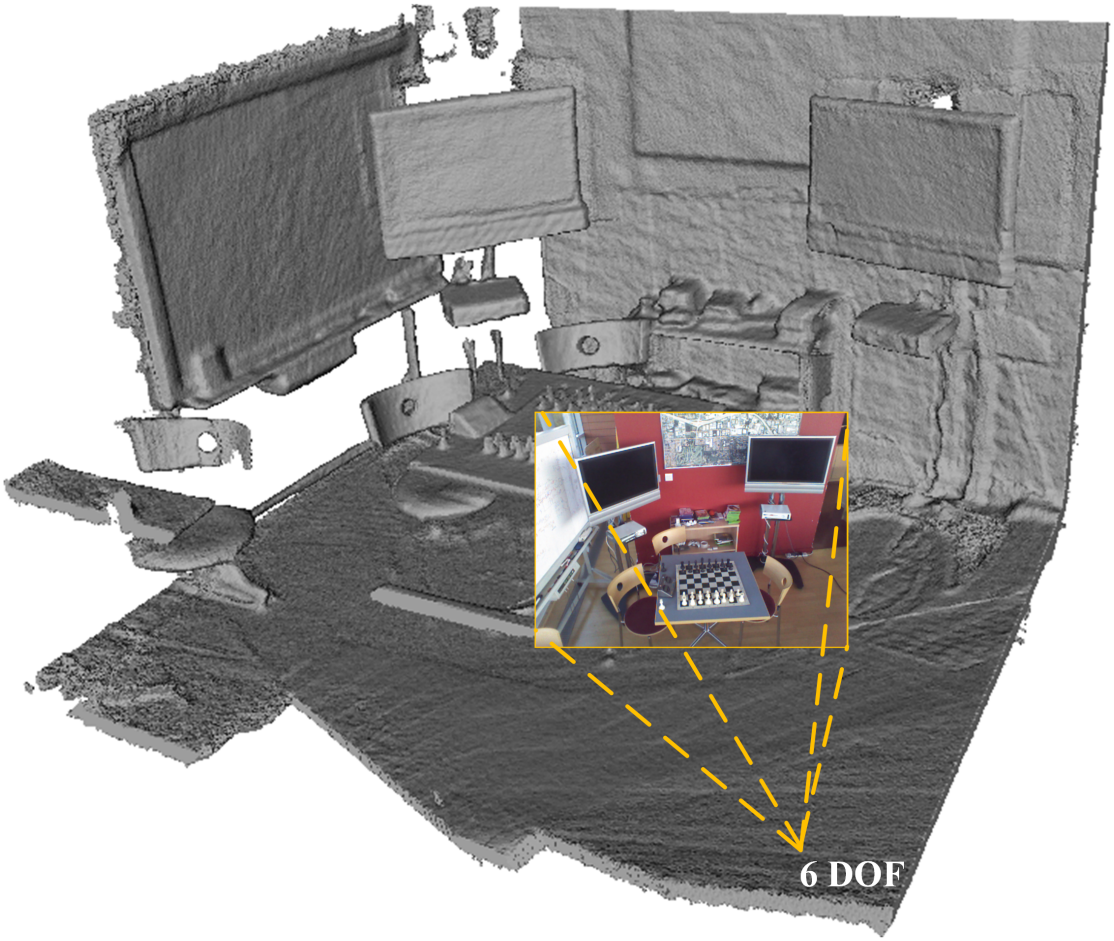


Figure 5.2: Demonstration of single indoor image localization in a 3D map.

extract a local map to perform geometry match instead of the global map for efficiency reason. The local 3D map is also utilized to generate the initial depth, and the initial depth is used to perform dense depth prediction with RGB image. The point cloud generation produces a point cloud with the coarse pose and the dense depth image. Eventually we exploit the ICP matching strategy to align the generated point cloud into the local 3D map to obtain the pose correction. By adding the correction to the initial pose, we obtain the accurate pose in the 3D maps. In the rest of this section, we describe the details of each step.

5.2.1 Pose Initialization

Pose initialization is the key component of the proposed approach. It provides the initial guess to extract the local 3D map from the global one and the ICP algorithm heavily relies on it to achieve good results. In this step, we utilize our previous approach in Chapter 4 to initialize the pose of the image, which can also be replaced with other localization methods. The pose initialization approach is also a learning-based

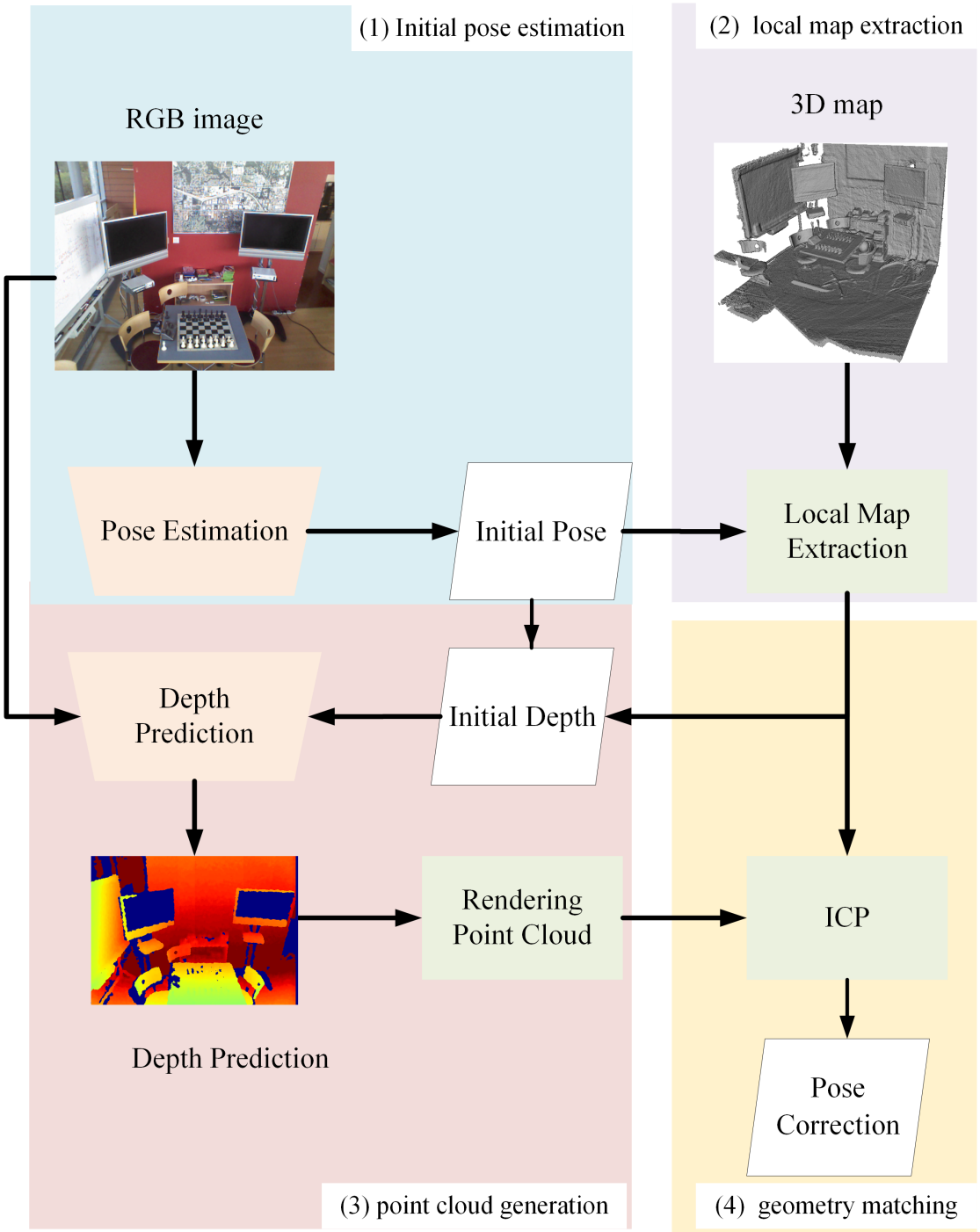


Figure 5.3: Image localization process. It includes four stages: (1) initial pose estimation, (2) local map extraction, (3) point cloud generation, (4) geometry matching.

method. A Siamese neural network structure is designed to exploit the relative geometry between images in both feature space and label space. The network consists of two shared-weighted ResNet50, two global pose regression units and a relative pose regression unit, which are made of three fully connected layers. Three loss functions

are designed in conjunction with the global pose loss function to train the network. The proposed network is capable of estimating the global poses and relative poses of two images. In addition, it can also predict a single image by feeding the image into one branch of the network. This strategy performs multi-task learning and adding relative geometrical constraints to regularize the network. It is capable of performing localization in low texture indoor environments. Although the localization accuracy is not as high as traditional 3D model-based methods, it is enough to generate a coarse pose estimate which will be refined in later steps.

5.2.2 Local 3D Map Extraction

The global map contains a large number of points, and matching against the global map is quite inefficient because many of which are not necessary as only a small portion of it is seen in the field of view of the query images. To increase the efficiency of warping 3D point cloud and ICP matching, we extract a local 3D map based on the initial pose. In [183], they generate the 3D local map by choosing points within a distance threshold to the initial position. Many points that are out of view still appear in the map, which results in low efficiency. Points of long distance are filtered, which are important for the further localization. Therefore, we propose an approach based on the image field of view that is able to avoid the problems. Given the initial pose, we calculate the angle of the global points in the polar coordinates system. Points within an angular window are selected as local 3D maps. The size of the window is determined based on the initial pose and image size. Given the image intrinsic parameters, the field of view (FOV) can be computed using equations (5.2.1) and (5.2.2).

$$fov_h = \arctan \frac{w}{2 \times f}, \tag{5.2.1}$$

$$fov_v = \arctan \frac{h}{2 \times f}, \tag{5.2.2}$$

where fov_h and fov_v represent the horizontal and vertical view of the camera, f is the focal length, and w, h are the width and height of the image. To include points that appear in the camera view on the local map, we set the window larger than the camera FOV size. Empirically, 15° on both vertical and horizontal works well on all the experiment settings.

5.2.3 Point Cloud Generation

We estimate the corresponding point cloud of the query image with two steps. Firstly, we predict the depth image of the query image using the approach proposed in section 5.3 and generate its corresponding point cloud with equation (5.3.1). Secondly, we filter point clouds based on the density distribution of indoor 3D points. The filtering is essential since not all depth values are predicted accurately, which will affect the final results of geometry in 3D space. The depth values of large errors exhibit as the floating points in the 3D space. To eliminate them, we use a simple point cloud filtering strategy, which is based on the number of points within the given radius. Let N_i denotes the number of points near point i, T represents the threshold, R denotes the given radius. If $N_i < T$, then the point i is reserved, otherwise the point i is abandoned. In our approach, the radius R is set to $1m$ and the point count threshold T is 100.

5.2.4 ICP-based Geometry Matching

Given the local 3D map and the predicted point cloud from an image, we perform the ICP algorithm [184] to align them. Like other point cloud alignment algorithms, the correspondences between the local map and generated point cloud are searched. Instead of building hard associations between two point clouds, the ICP approach updates the data associations in each iteration. For each iteration, the corresponding map points of the estimated point cloud are assigned with the nearest points in the local map. We select the point pairs with distance lower than the given threshold to compute the pose correction. The alignment result is added to the initial pose to obtain the final localization results.

The key to the above single image localization approach is estimating the depth image of the 2D RGB image. In the next Section, we will present a new approach that fuse the RGB image with the 3D map information to estimate the depth image of the RGB image.

5.3 Single Image Depth Prediction with 3D Map Guidance

Traditional single image depth prediction methods are performed directly from RGB images, and suffer from the scale ambiguity problem. To relieve it, we exploit the 3D maps to guide the process. The 3D maps information is utilized by generating an initial depth image of the input RGB images. Then both the RGB image and the initial depth image are fed into network to infer the correct depth image. In the rest of this section,

we elaborate on the details of warping 3D map to provide the initial depth image, and we also describe the architecture of the proposed convolutional neural network for depth prediction and the formulation of loss functions.

5.3.1 Initial Depth Generation

Initial depth images are generated by projecting the local 3D points into a plane that is defined by the camera intrinsic parameters and coarse pose. The camera position determines the camera plane position, and the camera orientation determines the normal of the camera plane. Depth image size and camera focal length determine the field of view (FOV) of the camera. According to the principle of the camera geometric projection, each pixel corresponds to a 3D point. More specifically, the 3D coordinates and the 2D corresponding points obey the pinhole camera geometry that can be expressed by equation (5.3.1):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{i,j} = R^{-1} \times D_{i,j} \times K^{-1} \times [i, j, 1] + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{cam}}, \tag{5.3.1}$$

where R represents the rotation matrix. $K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ is the intrinsic matrix, (u_0, v_0) is the principal point in the camera plane. The pixel value indicates the distance between the 3d points and the camera position. The image coordinate i, j on the depth image of a 3D points is jointly determined by camera position, orientation, depth image size, and camera focal length. The value of the depth image is the distance between the 3D points and the camera position, which can be computed with the equation (5.3.2):

$$D_{i,j} = \left\| \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{i,j} - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{cam}} \right\|, \tag{5.3.2}$$

where $D_{i,j}$ denotes the depth value at position (i, j) of the projecting plane, $\|*\|$ represents the Euclid distance, $\begin{bmatrix} X, Y, Z \end{bmatrix}^T$ and $\begin{bmatrix} X, Y, Z \end{bmatrix}^T_{\text{cam}}$ denote the 3D point and the camera position in the 3D map respectively.

Their position (i, j) on the depth map can be computed with the equation (5.3.3):

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = K \times \begin{bmatrix} R & T \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{5.3.3}$$

where $T = \begin{bmatrix} X, Y, Z, 1 \end{bmatrix}^T_{\text{cam}}$ represents the camera position in the homogeneous coordinate system.

We compute the corresponding depth image position of every points from the local 3D maps for efficiency reason. Positions out of the depth images are abandoned and the positions without the corresponding 3D points are filled with zeros.

5.3.2 Depth Prediction Network

Network Architecture. The network architecture is illustrated in Figure 5.4. Our network structure is designed based on [172] which achieved state-of-the-art results in depth prediction from RGB images. The network is composed of two components: an encoder and a decoder. We use the modified ResNet50 to encode the image information for the NYU-Depth-v2. For the KITTI dataset, we use the ResNet18 because of GPU limitation, since the size of the KITTI images is too large to process with ResNet50 encoder. We modified ResNet by replacing the last pooling layer and the fully connected layer with a convolutional layer and a batch normalization layer. We fuse the initial depth images and RGB images after the first convolutional layer. The decoder consists of four successive $uppool$ up-sampling layers and a bilinear up-sampling layer.

Loss Function . The loss function is designed based on the difference between the estimated depth image and the ground truth. Three common depth-wise losses are exploited for training the network, i.e. the mean squared error (l_2), the mean absolute error (l_1), and the reversed Huber loss $(berHu)$ [215]. The $BerHu$ loss can be seen as a compromise of l_2 and l_1 as it is equivalent to the l_1, and otherwise is approximate to l_2 . It is defined as:

$$\mathcal{L}_{berHu}(err) = \begin{cases} |err|, & \text{if } |err| \leq c \\ \frac{err^2+c^2}{2c}, & \text{otherwise} \end{cases} \tag{5.3.4}$$

where c is a parameter that is computed as 20% of the rank of the absolute depth error of all pixels for a batch, err represents the absolute depth error .

Depth-wise losses lead to smooth boundaries. Therefore, we further test gradient loss and SSIM loss [216] to keep the boundaries sharp. The SSIM loss aims to constrain the

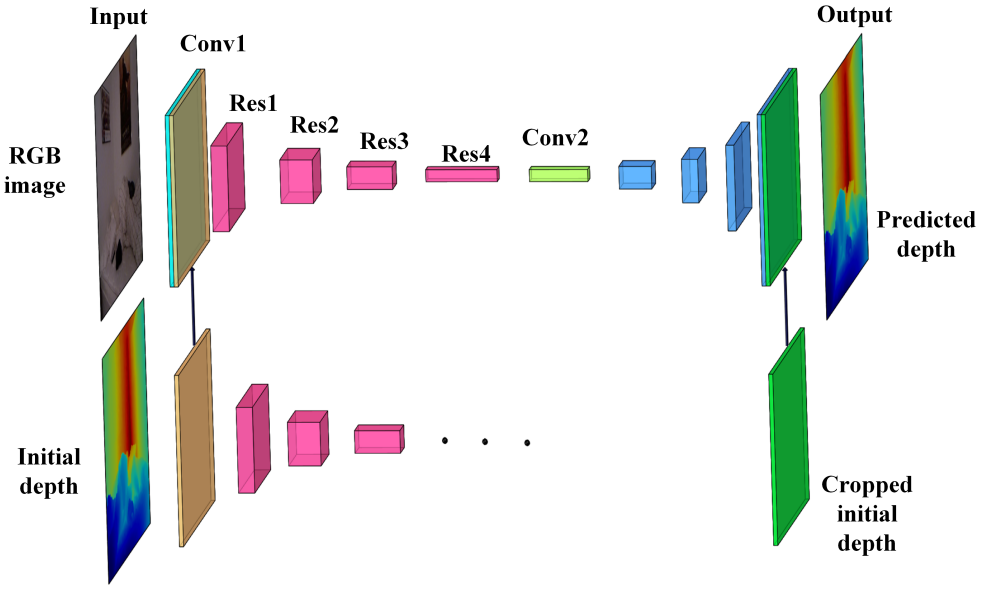


Figure 5.4: The architecture of the proposed network. The red blocks are the feature maps of residual blocks in ResNet, and blue blocks indicate the feature maps of upconv up-sampling layers. The green block is the cropped initial depth image which is concatenated with the RGB image.

difference of the predicted depth and the ground truth in appearance from a whole image. The SSIM loss is formulated as in equation (5.3.5).

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}, \tag{5.3.5}$$

where μ_x, μ_y are the average depth of images, the σ represents the standard deviation of two images, and C_1, C_2 are constant values, which equal to $0.01^2, 0.03^2$ respectively. The gradient loss is defined as in equation (5.3.6), which try to preserve detail information on the depth images.

$$L_g = \left\|dx - \tilde{dx}\right\| + \left\|dy - \tilde{dy}\right\|, \tag{5.3.6}$$

where the L_g denotes the gradient loss, dx, dy are the gradients computed from the ground truth depth image in x, y directions and $\tilde{dx}, \tilde{dy}$ are the gradients of the predicted depth image. The pixel-wise gradient loss encourages the gradient of the predicted depth image to be consistent with the ground-truth depth image. We will do experiments to compare the performance of different loss functions.

5.4 Experiments

In this section, we evaluate the performance of depth prediction and image localization respectively. We compare the depth estimation performance with the state-of-the-art on two benchmarks: the NYU-Depth-v2 [217] and the KITTI dataset [218]. Ablation studies are conducted on the network structure and loss functions. For localization, we evaluate our method on the *7Scene* dataset [128].

5.4.1 Depth Prediction

Datasets. The NYU-Depth-v2 dataset is collected from 464 different scenes with a Kinect device. It is officially split into training an testing dataset, where 249 scenes are selected for training and 215 scenes for testing. To facilitate comparison with the previous methods, we also evaluate our method on 654 images as in previous works [219–222]. Following previous work [172], we resize the image into 320×240 pixels and crop a patch of 304×228 pixels from the centre. The KITTI dataset is collected on a mobile car and the depth is obtained using a Velodyne LiDAR sensor. We use the split proposed by [221] in which 22,600 images are used for training and 697 images for testing. Only the bottom crop (912×228 pixels) is performed to eliminate the sky, where no depth information is acquired by the sensor.

Since both datasets have no accurate pose information, we simulate the initial pose for them. Three random numbers within $[-3t, 3t]$ are used to simulate initial position and three random numbers within $[-3\theta, 3\theta]$ act as the initial orientation, where θ and t are the median position error and orientation error in previous chapter [214]. For the NYU-Depth-v2 dataset, the t and θ are $0.2m$ and 10° respectively, according to the localization results on indoor scenes. For the KITTI dataset, the t and θ are $1.2m$ and 3.2° according to the performance in our outdoor localization results.

Setup. We follow the same data augmentation strategy as in [172] by random transformation on scale, rotation, colour, and flips on RGB images. Training images are shuffled before they are fed to the network. We choose Adam optimizer to train the network with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The weight decay is 1×10^{-5}. We train the network with a learning rate of 1×10^{-4} and the batch size of 12. We implement the network with PyTorch and train the network on an Ubuntu 16.04 LTS system with a NVIDIA GTX 1080Ti GPU. Training is stopped until the network is converged.

Evaluation Metrics. We evaluate the performance of current depth prediction with the following metrics:

The root mean square error (rmse):

$$rmse = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{d}_i - d_i\right)^2}. \tag{5.4.1}$$

The mean relative error (rel):

$$rel = \frac{1}{N}\sum_{i=1}^{N}\frac{\left\|\hat{d}_i - d_i\right\|}{d_i}. \tag{5.4.2}$$

The percentage of the relative depth prediction within threshold 1.25^j:

$$\delta_j = \frac{\mathrm{N}\left(\hat{d}_i : \max\left\{\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i}\right\} < 1.25^j\right)}{\mathrm{N}\left(d_i\right)}, \tag{5.4.3}$$

where d_i and $\hat{d}_i$ are the ground truth depth values and the predicted ones respectively, and N is the number of element of a set, $j = 1, 2, 3$. A higher δ_i indicates better prediction.

Comparison with State-of-the-art. We compare with existing monocular image depth prediction methods [172, 173, 219–226] on the depth prediction performance on NYU-Depth-v2 [217] and the KITTI dataset [218]. Among them, [173] and [172] also utilize the depth image whilst all other approaches only use the RGB image. The comparative results are shown in Table 5.1.

It can be seen from Table 5.1 that compared to RGB-based methods, RGBD-based methods achieve better performance on both error and accuracy. It is because RGB images only contain the relative distance information between pixels in them, and they need a target to convert the relative depth to their absolute corresponding depth values. RGBD-based methods jointly utilize the texture information from RGB images and absolute scale information from additional depth information, thus obtaining better results. Some qualitative examples are shown in Figure 5.5. It also is seen that RGB images can be used for relative depth estimation as the structure can be seen from the predicted depth images. RGBD-based results are more accurate as their pixel value is closer to that of the real depth image. By comparing RGBD-based methods, our method achieves comparable performance with Ma et al. [172] and outperforms the approach in [173]. It demonstrates the effectiveness of our proposed method. Although our depth map is not accurate, the initial depth is very dense and only has slight errors, which helps to enhance the depth prediction results.

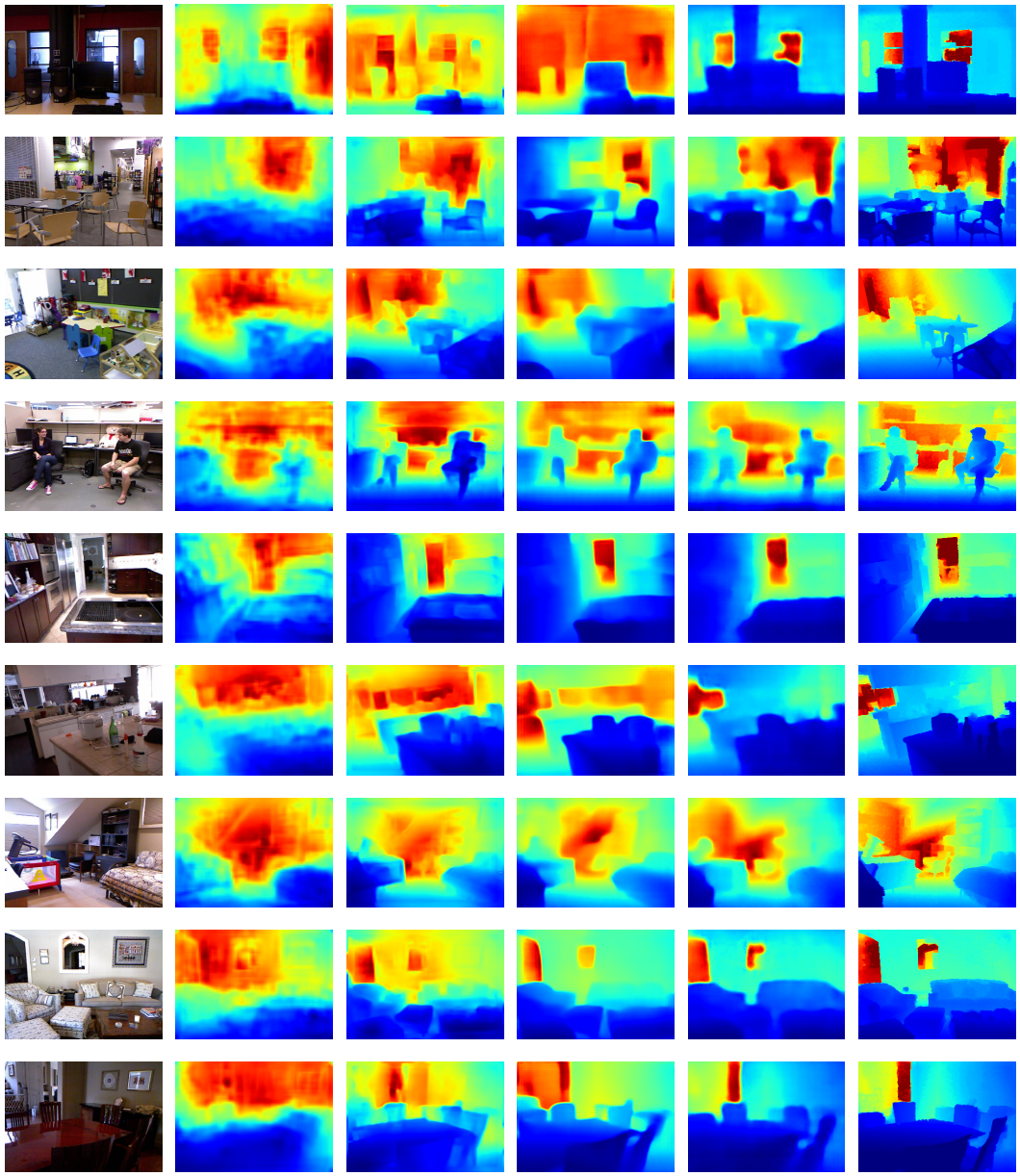
The KITTI dataset is more challenging compared to NYU-Depth-v2 dataset because it has larger distance up to $(100m)$ than that $(10m)$ of the NYU-Depth-v2 dataset. Besides,

Table 5.1: Comparison with the state-of-the-art on the NYU-Depth-v2 dataset. The reported values are referred to their papers respectively. The best performance is highlighted in bold.

Problem	Method	Error (lower is better)		Accuracy (higher is better)		
		rmse	rel	$\delta 1$	$\delta 2$	$\delta 3$
RGB	Karsch et al. [219]	1.200	0.250	-	-	-
	Liu et al. [220]	1.060	0.335	-	-	-
	Li et al. [223]	0.821	0.232	62.1	88.6	96.8
	Roy et al. [224]	0.744	0.187	-	-	-
	Liu et al. [225]	0.824	0.230	61.4	88.3	97.1
	Eigen2014 et al. [221]	0.877	0.214	61.4	88.8	97.1
	Eigen2015 et al. [222]	0.641	0.158	76.9	95.0	98.8
	Laina et al. [215]	0.573	0.127	81.1	95.3	98.8
	Xu et al. [226]	0.586	0.121	81.1	95.4	98.7
RGBD	Liao et al. [173]	0.442	0.104	87.8	96.4	98.9
	Ma et al. [172]	0.230	**0.044**	**97.1**	**99.4**	**99.8**
	Ours	**0.225**	0.070	94.9	99.1	99.7

Table 5.2: Comparison with the state-of-the-art on the KITTI dataset. The reported values are referred to their respective papers. The best performance is highlighted in bold.

Problem	Methods	Error (lower is better)		Accuracy (higher is better)		
		rmse	rel	$\delta 1$	$\delta 2$	$\delta 3$
RGB	Liu et al. [225]	6.986	0.217	64.7	88.2	96.1
	Eigen et al. [221]	6.179	0.197	69.2	89.9	96.7
	Cao et al. [227]	4.712	0.115	88.7	96.3	98.2
	Garg et al. [168]	5.104	0.169	74.0	90.4	96.2
	Godard et al. [169]	5.381	0.126	84.3	94.1	97.2
	Zhang et al. [175]	4.310	0.136	83.3	95.7	98.7
RGBD	Ma et al. [172]	3.378	0.073	93.5	97.6	98.9
	Liao et al. [173]	4.50	0.113	87.4	96.0	98.4
	Ours	**2.710**	**0.068**	**95.1**	**98.3**	**99.3**



(a) RGB **(b)** Eigen2014 **(c)** Eigen2015 **(d)** Laina **(e)** Ours **(f)** GroundTruth

Figure 5.5: Qualitative depth prediction results on the NYU-Depth-v2 dataset. The first column shows RGB images and column (b)-(d) are the results of similar methods, and the results of the proposed method are shown in column (e), and (f) shows the real depth images.

it was collected in outdoor environments, the scene geometry of which is complex and challenging due to plants and shallows. We also compare both RGB-based methods and RGBD-based methods for the KITTI dataset.

A similar conclusion can be drawn from Table 5.2 that RGBD-based methods achieve significantly better performance than RGB-based methods in the outdoor environment by comparing the results of group RGB and group RGBD. Comparing with the other two RGBD-based methods, our approach obtains a better performance. Compared to the NYU-Depth-v2 images, the images of the KITTI dataset are larger and the depth maximum is larger. Sparsely labelled depth images need more real depth values to achieve better results. Our initial depth images projected from LiDAR data are relatively denser to work that is better for large image and large scenes. We also give some the qualitative prediction results in Figure 5.6, which demonstrates that our method can effectively infer the depth of the RGB images.

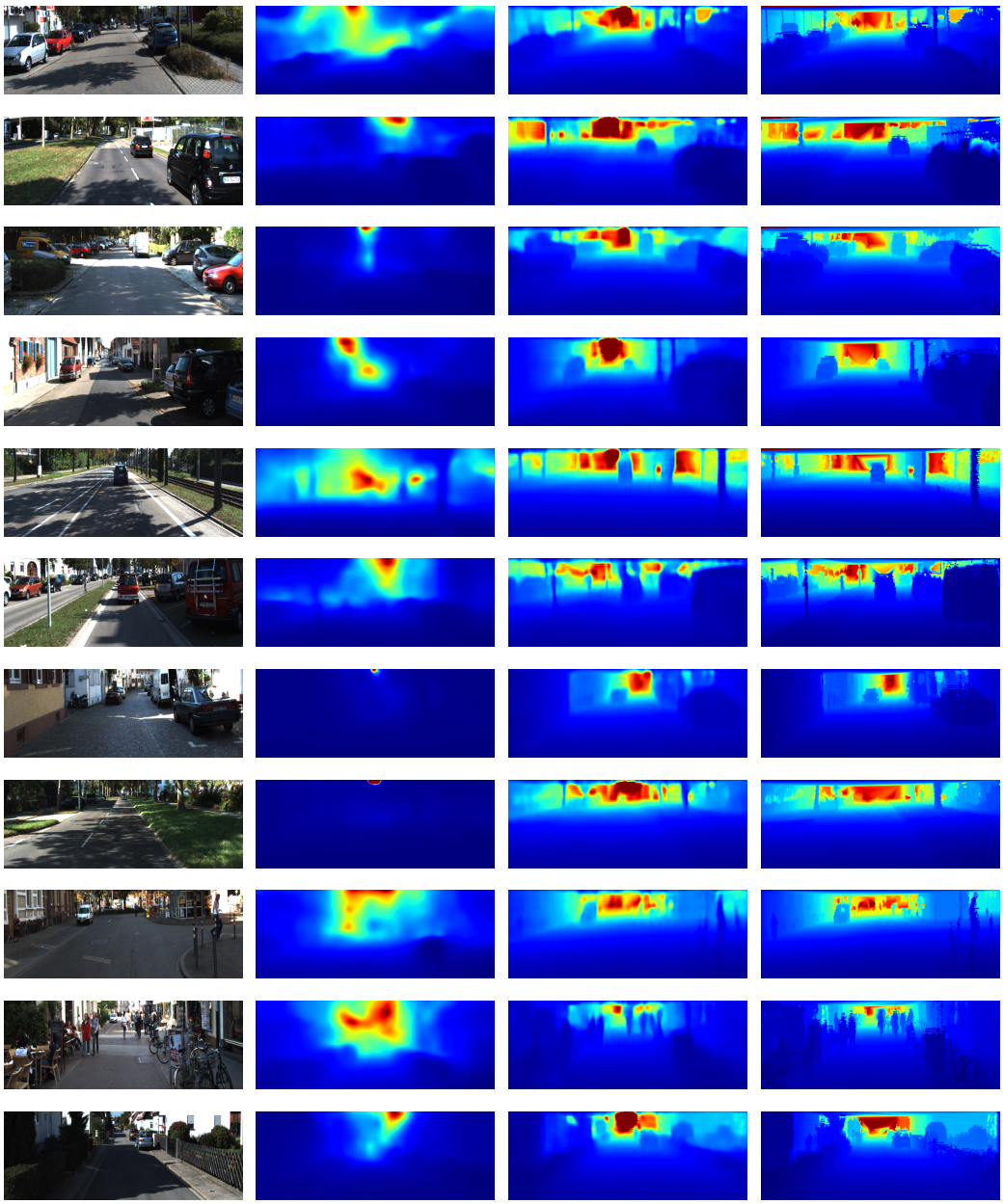
Analysis of Taking Different Input Data. We compare the depth prediction results of three kinds of input data including initial depth map (I), RGB images (RGB), and RGB images with their corresponding initial depth map $(RGBI)$. For I and RGB, the input channels are 1 and 3 respectively. For $RGBI$, the output of the $Conv1$ layer from I and RGB are concatenated for depth prediction. The rest of the networks are the same. The setup is the same with that in 5.4.1. The results are listed in Table 5.3 and 5.4, respectively.

Table 5.3: Results of different input data on the NYU-Depth-v2 dataset.

Input	Error (lower is better)		Accuracy (higher is better)		
	rmse	rel	$\delta 1$	$\delta 2$	$\delta 3$
I	0.320	0.108	89.1	98.4	99.7
RGB	0.514	0.143	81.0	95.9	98.9
RGBI	**0.228**	**0.070**	**94.3**	**98.9**	**99.8**

The results of I in Table 5.3 demonstrate that the proposed method is able to correct the depth value prediction from the inaccurate depth maps. The probable reason is that the structure of depth information is learned by the convolutional neural network after properly trained. Better results can be obtained if it is fused with RGB images as $RGBI$ contains both the correct structure information and global scale information.

For the KITTI dataset, the depth prediction results from initial depth alone are significantly worse than that from other input data. It is because unlike the NYU-Depth-v2, the ground truth depth for training network is sparse. Structure information is not



(a) RGB **(b)** Eigen2014 **(c)** Ours **(d)** GroundTruth

Figure 5.6: Qualitative depth prediction results of the KITTI dataset. (a) RGB images; (b) prediction results of Eigen ; (c) prediction of our method; (d) ground truth depth images.

Table 5.4: Results of different input data on the KITTI dataset.

Input	Error (lower is better)		Accuracy (higher is better)		
	rmse	rel	$\delta 1$	$\delta 2$	$\delta 3$
I	16.38	0.56	6.6	24.4	41.0
RGB	6.33	0.21	53.7	92.5	97.8
RGBI	**2.71**	**0.068**	**95.1**	**98.4**	**99.3**

able to be inferred by the network with sparse distributed ground truth depth values. However, although the initial map alone is not able to restore the dense depth, it is still capable of improving the results compared to results from RGB image alone.

Loss Analysis.

To find the proper loss function to train the network, we conduct ablation experiments to validate the effectiveness of three depth-wise losses and gradient loss, and SSIM loss. The experimental setup is repeated for each experiment as in section 5.4.1 except the loss function. To compare the loss function, we use the same network architecture in which the initial depth and RGB image information are concatenated after the first convolutional layer. We conducted the experiments on the NYU-Depth-v2 dataset. l_1, l_2, and $berHu$ loss are used to perform the comparison as well as their combination with the gradient loss and the SSIM loss. For the KITTI dataset, we only evaluate the l_1, l_2, and $berHu$ losses. It is because the ground truth depth is sparsely distributed in the KITTI dataset while the prediction is dense. Thus the SSIM loss and the gradient loss do not help in this case. The results are listed in Table 5.5 and Table 5.6 respectively.

To obtain the best localization accuracy, we pay more attention to the RSME since it represents the average absolute difference between the predicted depth values and the real ones.

In Table 5.5, it can be seen that l_1, l_2, and $berHu$ loss alone achieve good results and l_2 loss gets a slight better performance than the other two. We can also see that the gradient loss and the SSIM loss fails to work as performance decreases by adding them to the depth-wise loss. The reason might be that further loss makes the networks over-fit to the training data and decrease the generality to the testing data.

It can be seen from Table 5.6 that l_1, l_2, and $berHu$ losses have similar performance and the l_2 achieves slightly better performance than the other two on the *rmse*. As for localization, the *rmse* value is more important. Thus, we choose the l_2 to train our network.

Analysis of Fusion Strategy. To find the best fusion strategy of the initial depth image

Table 5.5: Evaluations of loss functions on the NYU-Depth-v2 dataset.

	Additional loss	Error (lower is better)		Accuracy (higher is better)		
		rmse	rel	$\delta 1$	$\delta 2$	$\delta 3$
l_1	+SSIM Loss	0.304	0.091	95.6	97.7	99.9
	+Gtradient Loss	0.275	0.092	91.5	98.7	99.8
	-	0.228	**0.070**	94.3	98.9	99.8
berHu	+SSIM Loss	0.598	0.153	80.5	97.6	99.3
	+Gtradient Loss	0.323	0.109	90.3	97.5	99.6
	-	0.243	0.075	94.1	98.9	99.8
l_2	+SSIM Loss	0.327	0.097	93.2	98.9	99.8
	+Gtradient Loss	0.303	0.104	90.2	98.0	99.7
	-	**0.225**	**0.070**	**94.9**	**99.1**	**99.8**

Table 5.6: Comparison of loss functions on the KITTI dataset.

Loss	Error (lower is better)		Accuracy (higher is better)		
	rmse	rel	$\delta 1$	$\delta 2$	$\delta 3$
l_1	2.73	**0.057**	**95.9**	98.3	99.1
berHu	2.74	0.058	95.8	98.2	99.2
l_2	**2.71**	0.068	95.1	**98.4**	**99.3**

and the RGB image, we conduct experiments to fuse the initial depth and RGB data in different layers, which are listed as below.

1. Input : Initial depth image and RGB image are concatenated before feeding into the network.
2. Conv1 : Initial depth image and RGB image are fused after the *convl* layer.
3. Res1 : Initial depth image and RGB image are fused after the *Res*1 block.
4. Res2 : Initial depth image and RGB image are fused after the *Res*2 block.
5. Res3 : Initial depth image and RGB image are fused after the *Res*3 block.
6. Output : Initial depth image and RGB image are fused before the last convolutional layer.

We use the same training setup in section 5.4.1 on NYU-Depth-v2 dataset. The training loss is l_1. The results are shown in Table 5.7.

Table 5.7: Evaluation of different fusion strategies on the NYU-Depth-v2 dataset.

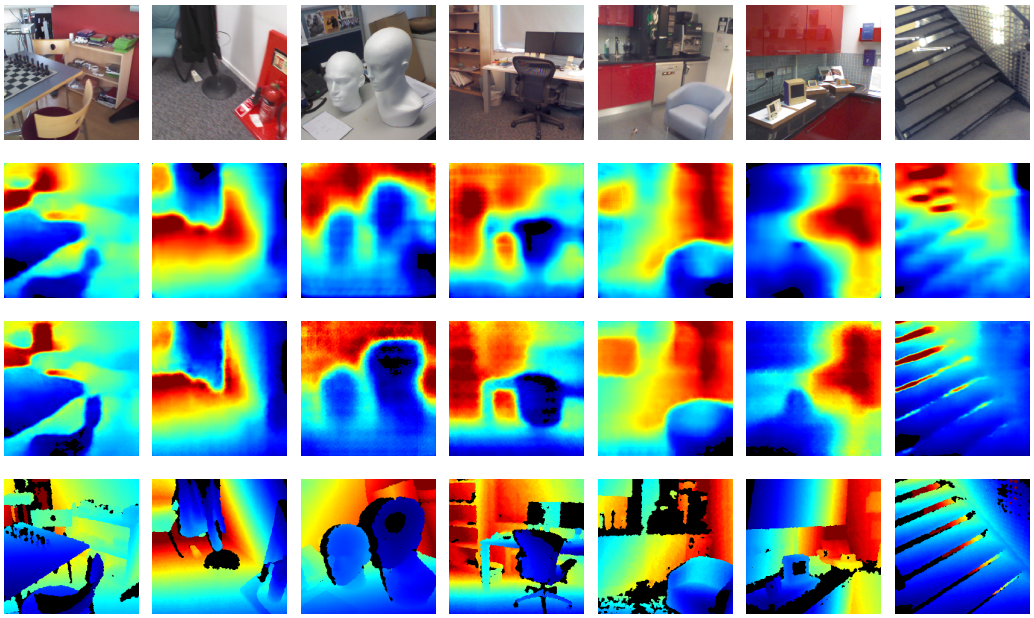
architecture	Error (lower is better)		Accuracy (higher is better)		
	rmse	rel	$\delta 1$	$\delta 2$	$\delta 3$
Input	0.231	**0.066**	**96.2**	**99.3**	99.8
Conv1	**0.228**	0.070	94.3	98.9	99.8
Res1	0.242	0.075	94.2	99.2	**99.9**
Res2	0.297	0.107	89.4	97.1	99.5
Res3	0.400	0.134	84.9	96.9	99.5
Output	0.600	0.191	71.8	93.0	98.0

It can be drawn from Table 5.7 that fusion in later layers gives worse results. In general the later the layer is, the worse the results are. The best performance is achieved in *Input* and *Conv*1 layers. Earlier network layers contain the spatial information, which is highly related to depth prediction. To fully maintain the spatial information, we also conduct experiment by adding initial depth information before last depth prediction layer, the results is the worst, which implies that the network requires the complex operation to fuse two sources information to get the best results.

5.4.2 Localization

Datasets. *7Scene* [128] is used to evaluate the performance of the localization. *7Scene* is an indoor image dataset for camera relocalization and trajectory tracking. It is collected with Kinect in a handhold manner. The ground truth pose and the 3D maps are generated using the Kinect Fusion approach [207]. The dataset is captured in 7 indoor scenes. For each scene, it contains several image sequences as well as the depth images sequence, which has already been divided into training and testing sets. The images are taken at the resolution of 640×480 pixels with the known the intrinsic parameters. We use officially split of the training set and testing set to train and test our depth estimation network.

Depth Prediction. We use the same setup for depth prediction of the *7Scene* dataset as in the NYU-Depth-v2 and perform the depth prediction from RGB images and RGBI data. The initial map is generated from the corresponding depth image by adding the certain position and orientation noise to the real pose as we did in the experiments of the NYU-Depth-v2. The l_2 loss is used to train the network. The results are shown in Table 5.8.



(a) chess **(b)** fire **(c)** heads **(d)** office **(e)** pumpkin **(f)** redkitchen **(g)** stairs

Figure 5.7: Qualitative depth prediction on 7Scenes dataset. The top row represents the RGB data, the second row are the prediction results from RGB image, the third row represents the results from RGBI data and the bottom row are the ground truth depth.

Table 5.8: Depth prediction results of the 7Scene dataset.

Dataset	Input	Error (lower is better)		Accuracy (higher is better)		
		rmse	rel	$\delta 1$	$\delta 2$	$\delta 3$
chess	RGB	0.215	0.076	93.9	98.8	99.7
	RGBI	0.186	0.167	95.5	99.1	99.9
fire	RGB	0.108	0.048	97.5	99.5	99.9
	RGBI	0.140	0.070	95.2	99.4	99.9
heads	RGB	0.160	0.140	78.7	94.1	98.8
	RGBI	0.112	0.109	90.3	98.7	99.8
office	RGB	0.243	0.093	91.2	98.6	99.7
	RGBI	0.184	0.069	95.4	99.1	99.8
pumpkin	RGB	0.155	0.054	97.6	99.5	99.8
	RGBI	0.149	0.049	97.4	99.4	99.8
redkitchen	RGB	0.228	0.087	92.6	99.0	99.9
	RGBI	0.208	0.079	94.6	99.3	99.9
stairs	RGB	0.344	0.091	87.4	96.4	99.3
	RGBI	0.268	0.072	91.6	98.0	99.4

By comparing the depth prediction results on the *7Scene* from RGB and RGBI information, we find that RGBI achieves better performance, which further verifies our idea on more datasets. Some qualitative prediction examples of 7 scenes are listed in Figure 5.7. It shows that the RGBI-based method also help improve the details on the structure than using RGB data alone.

Given the predicted depth, The ICP algorithm is applied to perform localization as described in section 5.2. The overlap parameter is set as 0.7 to eliminate the depth prediction of large error. The max iteration is set to 30 and the threshold to stop the iteration is set to 0.01. The localization result is shown in Table 5.9. RGB-depth represents the method that the depth maps are directly estimated from the RGB images and used for 3D localizations. RGBI-depth represents the method that the depth maps are estimated from the RGB images for 3D localizations. The results are reported with the median error to facilitate the comparison.

Table 5.9: Comparison with the CNN-based localization over the 7Scene dataset. The best localization results are highlighted in bold.

Dataset	PoseNet2[138]		Relnet [140]		Our CNN[214]		RGB-depth		RGBI-depth (ours)	
	Orientation	Position	Orientation	Position	Orientation	Position	Orientation	Position	Orientation	Position
chess	4.48°	0.13m	6.46°	0.13m	5.19°	0.099m	2.99°	**0.07m**	**2.49°**	0.077m
fire	11.3°	0.27m	12.72°	0.26m	11.64°	0.253m	3.12°	0.07m	**1.22°**	**0.035m**
heads	13.0°	0.17m	12.34°	0.14m	13.20°	0.126m	16.77°	0.29m	**6.44°**	**0.140m**
office	5.55°	0.19m	7.35°	0.21m	7.71°	0.161m	4.94°	0.15m	**4.66°**	**0.141m**
pumpkin	4.75°	0.26m	6.35°	0.24m	6.61°	0.163m	4.60°	**0.15m**	**4.03°**	0.154m
redkitchen	5.35°	0.23m	8.03°	0.24m	8.24°	0.174m	2.86°	0.15m	**2.45°**	**0.086m**
stairs	12.4°	0.35m	11.82°	0.27m	13.13°	0.260m	7.56°	0.23m	**2.48°**	**0.078m**
average	8.12°	0.23m	9.30°	0.21m	9.39°	0.177m	6.12°	0.17m	**3.40°**	**0.102m**

By comparing CNN-based pose regression methods [138, 140, 214] and depth prediction-based method, we can draw a conclusion that depth prediction helps increase the pose localization performance in both position and orientation. The accurate depth prediction results provide better localization by comparing RGB depth prediction-based method and RGBI depth prediction-based method. compared with the results in the previous chapter, shown as our CNN The positional error decreases from 0.177m to 0.102m and the orientational error drops from 9.39° to 3.40°.

5.5 Concluding Remarks

Single image indoor localization in 3D map is very important for many applications. This chapter presents a new framework to localize single images in 3D maps through RGB images depth inference and matching them based on their geometry similarity in 3D space. Moreover, we propose a new depth prediction method by warping the

3D maps information into initial depth for depth prediction. The depth prediction results outperform the state-of-the-art. We also evaluate our localization approach on the *7Scene* dataset, the experimental results demonstrate the effectiveness of our method in enhancing the localization accuracy. In principle, our method can be equally applied to single outdoor image localization. In fact, we have tested the algorithm on the outdoor dataset. However, due to the difficulty in obtaining an accurate 3D map, the performance is not as good as that of the indoor images. Our future work will focus on applying the method to the outdoor scenario. ICP algorithm takes up the most of time for localization as it needs to establish the 3D correspondences between the predicted point cloud and the 3D map iteratively. Depth prediction and initial pose estimation barely cost time.

In the next chapter, we will summarize the contents of this thesis, and we will restrengthen our contributions and discuss certain limitations of our current approaches and give some advice for the further improvements.

CHAPTER 6

Concluding Remarks

In this thesis, we have studied image-based indoor localization problems from topological localization and metric localization perspectives. In this chapter, we will summarize the major contributions of the thesis, point out the limitations, and provide some suggestions for further improvements.

6.1 Main Contributions

Our first contribution is to propose an indoor visual topological localization framework, called Visual Landmark Sequence-based Indoor Localization (VLSIL), which exploits the semantic information and contextual information derived from the videos. In this framework, we present a new topological map representation, in which the indoor environment is divided into many regions based on the existing fixed objects, and each region is represented with the semantic information of the fixed objects in it. This new representation scheme is invariant to the light or slight view change as well as the environmental change, as it concerns more about the high-level information rather than the visual and geometric information. We also propose a powerful landmark detector relying on convolutional neural network, which has been demonstrated to be more effective than the detectors based on conventional machine learning techniques for extracting semantic information from videos. Besides, a novel localization algorithm is proposed through exploiting both the semantic information and contextual information to match the detected landmark sequences against the topological map. Experiments on two challenging indoor test-beds show that it can accurately perform localization and achieve better performance than hidden Markov model-based methods.

Our second contribution is to propose a new deep learning-based approach for image-

based metric localization by utilizing the relative geometric constraints between images. Learning-based image pose estimation methods have many advantages compared to the traditional image retrieval-based method and SfM-based methods. They are efficient and have low storage and computational requirements. Moreover, they can handle scenes of low texture. Many learning-based methods only focus on the difference between real global poses and the predicted ones, while the relative geometry between images are ignored, which can further regularize the network for global pose estimation. We present a novel relative geometry-aware Siamese neural network to enhance the performance of deep learning-based methods through explicitly exploiting the relative geometry constraints between images. We perform multi-task learning and predict the absolute and relative poses simultaneously. We regularize the shared-weight twin networks in both the pose and feature domains to ensure that the estimated poses are globally as well as locally correct. We employ metric learning and design a novel adaptive metric distance loss to learn a feature that is capable of distinguishing poses of visually similar images of different locations. We evaluate the proposed method on public indoor and outdoor benchmarks and the experimental results demonstrate that our method can significantly improve localization performance. Furthermore, extensive ablation evaluations are conducted to demonstrate the effectiveness of different terms of the loss function.

The third contribution is a single image metric localization in 3D maps framework. With the development of 3D sensors, many 3D models are built using them. 3D information is very helpful for image localization. However, such 3D information is seldom used for image localization because previous methods require the 3D information to be associated with visual local features while 3D models collected with LiDAR devices have no visual information. We propose a new framework to address it by matching geometry between a single image and a 3D map. The framework includes four main steps: pose initialization, local map extraction, depth prediction, and geometry matching. The depth prediction is the key for single image localization in a 3D map. Previous methods only generate sparse depth information from multiple overlapped images and the procedure is quite slow. We propose a new dense depth map estimation method by utilizing the convolutional neural network. Compared to other CNN-based depth prediction methods, which take only RGB images as input, our method predicts the depth map from an RGB image and an initial depth map generated from 3D maps. The new depth method is evaluated on both indoor and outdoor depth prediction datasets, and achieves the state-of-the-art results. The results on an indoor localization dataset demonstrate the effectiveness of the proposed method and the 3D maps can help improve the accuracy of image-based localization.

6.2 Limitations and Suggestions for Improvement

This thesis provides several new methods for indoor image localization using deep learning techniques.

For topological localization, the proposed approach has been demonstrated to be effective in the indoor scenes when landmarks are all correctly detected. Our localization algorithm is not able to handle the case that the landmarks are wrongly detected or missing in the landmark sequence. The probable reason is that our localization algorithm only considers the connecting information between adjacent nodes, and takes the landmark detection results for granted without considering the probability of wrong detection of the landmarks. The localization algorithm can be improved by incorporating the landmark detector confidence and the probability of the skipped connection of non-adjacent nodes.

For learning-based single image metric localization, we predict the single image pose by exploiting the relative geometric information between training images. We have to train a deep learning model for every single scene and a general model for the universal scenes is still an open problem. In our method, we exploit the visual constraints to further regularize the network. Traditional local point geometry constraints are not exploited to further regularize the networks. It is a promising direction to introduce photometric constraints to train the network.

In term of image metric localization with 3D map assistance, the key lies in generating an accurate depth map. We design a network structure in an encoder-decoder manner and the predicted depth image has smooth boundaries. The probable reason is that part of the image spatial structure information are abandoned due to down-sampling layers such as convolution and pooling operation. The structure information is extracted by the encoder layers of the network, thus designing an effective fusion strategy to predict the depth with the features of both encoder layers and decoder layers of the network may be capable of increasing the performance.

6.3 Summary

In summary, we have investigated indoor image-based localization problem through deep learning techniques on both topological localization and metric localization in the thesis. Topological localization aims to predict the coarse position while the metric localization tries to estimate the accurate pose, which consists of both position and orientation. We have developed a novel topological localization framework to conduct

localization with smart phone videos. It takes consideration of semantic information of fixed objects in the indoor environments. A novel topological map is presented, and a localization algorithm is devised to match the detected semantic information against the topological map.

However, for certain applications like robots navigation or autonomous driving, the accurate poses are needed. We employ the deep learning-based method to fast and accurately predict the pose. Although previous works [132, 138, 140] also adopt this strategy, they usually take the single image as input and ignore the relative geometry constraints between training images. In contrast, we design a loss function that considers the relative geometry between training samples to further constrain the network. Besides, we also perform multi-task learning to jointly predict the relative pose and global pose.

Since there are many available 3D models of indoor scenes, we develop a metric localization approach with a single image, which as far as we know is the first to perform single image localization in 3D maps. The method localizes the single image through geometry matching in 3D space by inferring the depth image from the RGB data. To address the scale ambiguity problem, we warp the 3D map to generate an initial depth image to guide the depth prediction, and experiments on indoor and outdoor benchmarks demonstrate that such depth prediction method is superior to the RGB-based methods.

Topological localization and metric localization are mutually beneficial solutions. Topological localization can divide the scene into many local maps, and thus perform metric localization in each local maps reduces the computational and storage requirements that metric localization suffers from. Besides, it can avoid the trouble caused by different locations of similar appearance. Metric localization results can improve the representation of the topological map as it can register many images taken under different conditions. We have made improvements in the two areas separately in this thesis. In the future, we will attempt to solve the problem simultaneously.

References

[1] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 165–178, 2016.

[2] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 483–494. ACM, 2014.

[3] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM computer communication review*, volume 45, pages 269–282. ACM, 2015.

[4] Marco Altini, Davide Brunelli, Elisabetta Farella, and Luca Benini. Bluetooth indoor localization with multiple neural networks. In *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, pages 295–300. IEEE, 2010.

[5] Lauri Aalto, Nicklas Göthlin, Jani Korhonen, and Timo Ojala. Bluetooth and wap push based location-aware mobile advertising system. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 49–58. ACM, 2004.

[6] Sinan Gezici, Zhi Tian, Georgios B Giannakis, Hisashi Kobayashi, Andreas F Molisch, H Vincent Poor, and Zafer Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE signal processing magazine*, 22(4):70–84, 2005.

[7] Sivanand Krishnan, Pankaj Sharma, Zhang Guoping, and Ong Hwee Woon. A uwb based localization system for indoor robot navigation. In *Ultra-Wideband, 2007. ICUWB 2007. IEEE International Conference on*, pages 77–82. IEEE, 2007.

[8] Lionel M Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P Patil. Landmarc: indoor location sensing using active rfid. In *Proceedings of the First IEEE Interna-*

tional Conference on Pervasive Computing and Communications, 2003.(PerCom 2003)., pages 407–415. IEEE, 2003.

[9] Nan Li and Burcin Becerik-Gerber. Performance-based evaluation of rfid-based indoor location sensing solutions for the built environment. *Advanced Engineering Informatics*, 25(3):535–546, 2011.

[10] Ricardo Tesoriero, José A Gallud, Manuel Lozano, and Victor M Ruiz Penichet. Using active and passive rfid technology to support indoor location-aware systems. *IEEE Transactions on Consumer Electronics*, 54(2):578–583, 2008.

[11] Kun-Chan Lan and Wen-Yuah Shih. On calibrating the sensor errors of a pdr-based indoor localization system. *Sensors*, 13(4):4781–4810, 2013.

[12] Samuel House, Sean Connell, Ian Milligan, Daniel Austin, Tamara L Hayes, and Patrick Chiang. Indoor localization using pedestrian dead reckoning updated with rfid-based fiducials. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 7598–7601. IEEE, 2011.

[13] Wonho Kang and Youngnam Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal*, 15(5):2906–2916, 2014.

[14] Benjamin Kuipers. Modeling spatial knowledge. *Cognitive science*, 2(2):129–153, 1978.

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

[16] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[17] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[18] D.G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1150–1157 vol.2, 1999. ISSN 0-7695-0164-8. doi: 10.1109/ICCV.1999.790410. URL `http://ieeexplore.ieee.org/document/790410/`.

[19] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, 2012.

[20] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[21] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[22] Xiaozhi Qu, Bahman Soheilian, Emmanuel Habets, and Nicolas Paparoditis. Evaluation of sift and surf for vision based localization. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41, 2016.

[23] Elena Stumm, Christopher Mei, Simon Lacroix, Juan Nieto, Marco Hutter, and Roland Siegwart. Robust visual place recognition with graph kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4535–4544, 2016.

[24] Youji Feng, Lixin Fan, and Yihong Wu. Fast localization in large-scale environments using supervised indexing of binary features. *IEEE Transactions on Image Processing*, 25(1):343–358, 2015.

[25] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 IEEE international conference on computer vision (ICCV)*, pages 2548–2555. Ieee, 2011.

[26] Clemens Arth, Christian Pirchheim, Jonathan Ventura, Dieter Schmalstieg, and Vincent Lepetit. Instant outdoor localization and slam initialization from 2.5 d maps. *IEEE transactions on visualization and computer graphics*, 21(11):1309–1318, 2015.

[27] Brittany Morago, Giang Bui, and Ye Duan. 2d matching using repetitive and salient features in architectural images. *IEEE Transactions on Image Processing*, 25 (10):4888–4899, 2016.

[28] Srikumar Ramalingam, Sofien Bouaziz, and Peter Sturm. Pose estimation using both points and lines for geo-localization. In *2011 IEEE International Conference on Robotics and Automation*, pages 4716–4723. IEEE, 2011.

[29] Bryan C Russell, Josef Sivic, Jean Ponce, and Helene Dessales. Automatic alignment of paintings and photographs depicting a 3d scene. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 545–552. IEEE, 2011.

[30] Shuda Li and Andrew Calway. Absolute pose estimation using multiple forms of correspondences from rgb-d frames. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4756–4761. IEEE, 2016.

[31] Eduardo Fernández-Moral, Walterio Mayol-Cuevas, Vicente Arévalo, and Javier Gonzalez-Jimenez. Fast place recognition with plane-based maps. In *2013 IEEE International Conference on Robotics and Automation*, pages 2719–2724. IEEE, 2013.

[32] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42 (3):145–175, 2001.

[33] Charbel Azzi, Daniel C Asmar, Adel H Fakih, and John S Zelek. Filtering 3d keypoints using gist for accurate image-based localization. In *BMVC*, 2016.

[34] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *2008 ieee conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[35] Peter Corke, Rohan Paul, Winston Churchill, and Paul Newman. Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2085–2092. IEEE, 2013.

[36] Andrew P Gee and Walterio W Mayol-Cuevas. 6d relocalisation for rgbd cameras using synthetic view regression. In *BMVC*, volume 1, page 2, 2012.

[37] Kai Ni, Anitha Kannan, Antonio Criminisi, and John Winn. Epitomic location recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(12): 2158–2167, 2009.

[38] Xue Wan, Jianguo Liu, Hongshi Yan, and Gareth LK Morgan. Illumination-invariant image matching for autonomous uav localisation based on optical sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 119:198–213, 2016.

[39] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 5297–5307, 2016.

[40] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017.

[41] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3251–3260. IEEE, 2017.

[42] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European conference on computer vision*, pages 3–20. Springer, 2016.

[43] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[44] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4297–4304. IEEE, 2015.

[45] Sung Joon Ahn, W Rauh, and M Recknagel. Circular coded landmark for optical 3d-measurement and robot vision. In *Ieee/rsj International Conference on Intelligent Robots and Systems, 1999. IROS '99. Proceedings*, pages 1128–1133 vol.2, 1999.

[46] Anahid Basiri, Pouria Amirian, and Adam Winstanley. The use of quick response (qr) codes in landmark-based pedestrian navigation. *International Journal of Navigation and Observation*, 2014, 2014.

[47] A. J. Briggs, D. Scharstein, D. Braziunas, C. Dima, and P. Wall. Mobile robot navigation using self-similar landmarks. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA*, pages 1428–1434 vol.2, 2002.

[48] Jean-Bernard Hayet, Frédéric Lerasle, and Michel Devy. A visual landmark framework for indoor mobile robot navigation. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 3942–3947. IEEE, 2002.

[49] V Ayala, J. B Hayet, F Lerasle, and M Devy. Visual localization of a mobile robot in indoor environments using planar landmarks. In *Ieee/rsj International Conference on Intelligent Robots and Systems*, pages 275–280 vol.1, 2000.

[50] YingLi Tian, Xiaodong Yang, Chucai Yi, and Aries Arditi. Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments. *Machine vision and applications*, 24(3):521–535, 2013.

[51] Kuan-Chieh Chen and Wen-Hsiang Tsai. Vision-based autonomous vehicle guidance for indoor security patrolling by a sift-based vehicle-localization technique. *IEEE transactions on vehicular technology*, 59(7):3261–3271, 2010.

[52] Yicheng Bai, Wenyan Jia, Hong Zhang, Zhi-Hong Mao, and Mingui Sun. Landmark-based indoor positioning for visually impaired individuals. In *Signal Processing (ICSP), 2014 12th International Conference on*, pages 668–671. IEEE, 2014.

[53] M Serrão, João MF Rodrigues, JI Rodrigues, and JM Hans du Buf. Indoor localization and navigation for blind persons using visual landmarks and a gis. *Procedia Computer Science*, 14:65–73, 2012.

[54] Hisato Kawaji, Koki Hatada, Toshihiko Yamasaki, and Kiyoharu Aizawa. Image-based indoor positioning system: fast image matching using omnidirectional panoramic images. In *Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis*, pages 1–4. ACM, 2010.

[55] Craig Becker, Joaquin Salas, Kentaro Tokusei, and J-C Latombe. Reliable navigation using landmarks. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 401–406. IEEE, 1995.

[56] Barbara Zitová and Jan Flusser. Landmark recognition using invariant features. *Pattern Recognition Letters*, 20(5):541–547, 1999.

[57] Andry Maykol G Pinto, A Paulo Moreira, and Paulo G Costa. Indoor localization system based on artificial landmarks and monocular vision. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 10(4):609–620, 2012.

[58] Guoyu Lin and Xu Chen. A robot indoor position and orientation method based on 2d barcode landmark. *JCP*, 6(6):1191–1197, 2011.

[59] Dimitrios I. Kosmopoulos and Konstantinos V. Chandrinos. *Definition and Extraction of Visual Landmarks for Indoor Robot Navigation*. Springer Berlin Heidelberg, 2002.

[60] Amir Roshan Zamir and Mubarak Shah. Accurate image localization based on google maps street view. In *European Conference on Computer Vision*, pages 255–268. Springer, 2010.

[61] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1546–1558, 2014.

[62] Elena Stumm, Christopher Mei, Simon Lacroix, and Margarita Chli. Location graphs for visual place recognition. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5475–5480. IEEE, 2015.

[63] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[64] Shervin Ardeshir, Amir Roshan Zamir, Alejandro Torroella, and Mubarak Shah. Gis-assisted object detection and geospatial localization. In *European Conference on Computer Vision*, pages 602–617. Springer, 2014.

[65] Michael Milford, Chunhua Shen, Stephanie Lowry, Niko Suenderhauf, Sareh Shirazi, Guosheng Lin, Fayao Liu, Edward Pepperell, Cesar Lerma, Ben Upcroft, et al. Sequence searching with deep-learnt depth for condition-and viewpoint-invariant route-based place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–25, 2015.

[66] Christian Poglitsch, Clemens Arth, Dieter Schmalstieg, and Jonathan Ventura. [poster] a particle filter approach to outdoor localization using image-based rendering. In *2015 IEEE International Symposium on Mixed and Augmented Reality*, pages 132–135. IEEE, 2015.

[67] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.

[68] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

[69] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2161–2168. Ieee, 2006.

[70] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[71] Mathieu Aubry, Bryan C Russell, and Josef Sivic. Painting-to-3d model alignment via discriminative visual elements. *ACM Transactions on Graphics (ToG)*, 33(2):14, 2014.

[72] Song Cao and Noah Snavely. Graph-based discriminative learning for location recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 700–707, 2013.

[73] Colin McManus, Ben Upcroft, and Paul Newmann. Scene signatures: Localised and point-less features for localisation. 2014.

[74] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Data-driven visual similarity for cross-domain image matching. In *ACM Transactions on Graphics (ToG)*, volume 30, page 154. ACM, 2011.

[75] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Predicting good features for image geo-localization using per-bundle vlad (open access). Technical report, University of North Carolina at Chapel Hill Chapel Hill United States, 2016.

[76] A Torralba, KP Murphy, WT Freeman, and MA Rubin. Context-based vision system for place and object recognition. In *Proceedings Ninth IEEE International Conference on Computer Vision*.

[77] Pilailuck Panphattarasap and Andrew Calway. Visual place recognition using landmark distribution descriptors. In *Asian Conference on Computer Vision*, pages 487–502. Springer, 2016.

[78] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.

[79] Yongliang Qiao, Cindy Cappelle, Yassine Ruichek, and Tao Yang. Convnet and lsh-based visual localization using localized sequence matching. *Sensors*, 19(11): 2439, 2019.

[80] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pages 392–407. Springer, 2014.

[81] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[82] Zhenghua Chen, Han Zou, Hao Jiang, Qingchang Zhu, Yeng Chai Soh, and Lihua Xie. Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization. *Sensors*, 15(1):715–732, 2015.

[83] Zhi-An Deng, Guofeng Wang, Danyang Qin, Zhenyu Na, Yang Cui, and Juan Chen. Continuous indoor positioning fusing wifi, smartphone sensors and landmarks. *Sensors*, 16(9):1427, 2016.

[84] Fuqiang Gu, Kourosh Khoshelham, Jianga Shang, and Fangwen Yu. Sensory landmarks for indoor localization. In *Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS), 2016 Fourth International Conference on*, pages 201–206. IEEE, 2016.

[85] Alexandra Millonig and Katja Schechtner. Developing landmark-based pedestrian-navigation systems. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):43–49, 2007.

[86] Margrit Betke and Leonid Gurvits. Mobile robot localization using landmarks. *IEEE transactions on robotics and automation*, 13(2):251–263, 1997.

[87] Beatriz L Boada, Dolores Blanco, and Luis Moreno. Symbolic place recognition in voronoi-based maps by using hidden markov models. *Journal of Intelligent & Robotic Systems*, 39(2):173–197, 2004.

[88] Baoding Zhou, Qingquan Li, Qingzhou Mao, Wei Tu, and Xing Zhang. Activity sequence-based indoor pedestrian localization using smartphones. *IEEE Transactions on Human-Machine Systems*, 45(5):562–574, 2015.

[89] Jana Kosecká and Fayin Li. Vision based topological markov localization. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1481–1486. IEEE, 2004.

[90] Martin Werner, Moritz Kessel, and Chadly Marouane. Indoor positioning using smartphone camera. In *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, pages 1–6. IEEE, 2011.

[91] Jason Zhi Liang, Nicholas Corso, Eric Turner, and Avideh Zakhor. Image based localization in indoor environments. In *Computing for Geospatial Research and Ap-*

plication (COM. Geo), 2013 Fourth International Conference on, pages 70–75. IEEE, 2013.

[92] Chi Chen, Bisheng Yang, Shuang Song, Mao Tian, Jianping Li, Wenxia Dai, and Lina Fang. Calibrate multiple consumer rgb-d cameras for low-cost and efficient 3d indoor mapping. *Remote Sensing*, 10(2):328, 2018.

[93] Pengcheng Zhao, Qingwu Hu, Shaohua Wang, Mingyao Ai, and Qingzhou Mao. Panoramic image and three-axis laser scanner integrated approach for indoor 3d mapping. *Remote Sensing*, 10(8):1269, 2018.

[94] Guoyu Lu and Chandra Kambhamettu. Image-based indoor localization system based on 3d sfm model. In *IS&T/SPIE Electronic Imaging*, pages 90250H–90250H. International Society for Optics and Photonics, 2014.

[95] Dominik Van Opdenbosch, Georg Schroth, Robert Huitl, Sebastian Hilsenbeck, Adrian Garcea, and Eckehard Steinbach. Camera-based indoor positioning using scalable streaming of compressed binary image signatures. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 2804–2808. IEEE, 2014.

[96] Harlan Hile and Gaetano Borriello. Positioning and orientation in indoor environments using camera phones. *IEEE Computer Graphics and Applications*, 28(4), 2008.

[97] Alessandro Mulloni, Daniel Wagner, Istvan Barakonyi, and Dieter Schmalstieg. Indoor positioning and navigation with camera phones. *IEEE Pervasive Computing*, 8(2), 2009.

[98] Guoyu Lu, Yan Yan, Nicu Sebe, and Chandra Kambhamettu. Indoor localization via multi-view images and videos. *Computer Vision and Image Understanding*, 2017.

[99] Guoyu Lu, Yan Yan, Li Ren, Philip Saponaro, Nicu Sebe, and Chandra Kambhamettu. Where am i in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging. *Neurocomputing*, 173:83–92, 2016.

[100] Claudio Piciarelli. Visual indoor localization in known environments. *IEEE Signal Processing Letters*, 23(10):1330–1334, 2016.

[101] Farhang Vedadi and Shahrokh Valaee. Automatic visual fingerprinting for indoor image-based localization applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[102] Nuri Lee, Changjae Kim, Wonseok Choi, Muwook Pyeon, and Yongil Kim. Development of indoor localization system using a mobile data acquisition platform and bow image matching. *KSCE Journal of Civil Engineering*, 21(1):418–430, 2017.

[103] Yannis Kalantidis, Giorgos Tolias, Yannis Avrithis, Marios Phinikettos, Evaggelos Spyrou, Phivos Mylonas, and Stefanos Kollias. Viral: Visual image retrieval and localization. *Multimedia Tools and Applications*, 51(2):555–592, 2011.

[104] Yong-Hwan Lee and Youngseop Kim. Efficient image retrieval using advanced surf and dcd on mobile platform. *Multimedia Tools and Applications*, 74(7):2289–2299, 2015.

[105] Xinchao Li, Martha Larson, and Alan Hanjalic. Geo-distinctive visual element matching for location estimation of images. *IEEE Transactions on Multimedia*, 20 (5):1179–1194, 2018.

[106] Ben JA Kröse, Nikos Vlassis, Roland Bunschoten, and Yoichi Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, 2001.

[107] Emanuele Menegatti, Mauro Zoccarato, Enrico Pagello, and Hiroshi Ishiguro. Image-based monte carlo localisation with omnidirectional images. *Robotics and Autonomous Systems*, 48(1):17–30, 2004.

[108] Junqiu Wang, Hongbin Zha, and Roberto Cipolla. Coarse-to-fine vision-based localization by indexing scale-invariant features. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(2):413–422, 2006.

[109] Junqiu Wang, Roberto Cipolla, and Hongbin Zha. Vision-based global localization using a visual vocabulary. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 4230–4235. IEEE, 2005.

[110] Akihiko Torii, Yafei Dong, Masatoshi Okutomi, Josef Sivic, and Tomas Pajdla. Efficient localization of panoramic images using tiled image descriptors. *Information and Media Technologies*, 9(3):351–355, 2014.

[111] Masataka Umeda and Hisashi Date. Spherical panoramic image-based localization by deep learning. *Transactions of the Society of Instrument and Control Engineers*, 54:483–493, 2018.

[112] Abner Guzman-Rivera, Pushmeet Kohli, Ben Glocker, Jamie Shotton, Toby Sharp, Andrew Fitzgibbon, and Shahram Izadi. Multi-output learning for cam-

era relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1114–1121, 2014.

[113] Iwan Ulrich and Illah Nourbakhsh. Appearance-based place recognition for topological localization. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1023–1029. Ieee, 2000.

[114] J. Kosecka, Liang Zhou, P. Barber, and Z. Duric. Qualitative image based localization in indoors environments. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, pages II–3–II–8 vol.2, 2003.

[115] Niko Sünderhauf and Peter Protzel. Brief-gist-closing the loop by simple means. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1234–1241. IEEE, 2011.

[116] Gautam Singh and J Kosecka. Visual loop closing using gist descriptors in manhattan world. In *ICRA Omnidirectional Vision Workshop*, 2010.

[117] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, J Javier Yebes, and Sergio Gámez. Bidirectional loop closure detection on panoramas for visual navigation. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 1378–1383. IEEE, 2014.

[118] Ana Cris Murillo and Jana Kosecka. Experiments in place recognition using gist panoramas. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 2196–2203. IEEE, 2009.

[119] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE, 2012.

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

[121] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[122] Michael Donoser and Dieter Schmalstieg. Discriminative feature-to-point matching in image-based localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 516–523, 2014.

[123] Torsten Sattler, Michal Havlena, Filip Radenovic, Konrad Schindler, and Marc Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2102–2110, 2015.

[124] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9):1744–1756, 2017.

[125] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *European conference on computer vision*, pages 752–765. Springer, 2012.

[126] M Uyttendaele, MF Cohen, SN Sinha, and Hyon Lim. Real-time image-based 6-dof localization in large-scale environments. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1043–1050. IEEE, 2012.

[127] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[128] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[129] Julien Valentin, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip HS Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4400–4408, 2015.

[130] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017.

[131] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016.

[132] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional

network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[133] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[134] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. *arXiv preprint arXiv:1509.05909*, 2015.

[135] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 870–877. IEEE, 2017.

[136] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *Int. Conf. Comput. Vis.(ICCV)*, pages 627–637, 2017.

[137] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017.

[138] Alex Kendall, Roberto Cipolla, et al. Geometric loss functions for camera pose regression with deep learning. In *Proc. CVPR*, volume 3, page 8, 2017.

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

[140] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 920–929. IEEE, 2017.

[141] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. *arXiv preprint arXiv:1803.03642*, 2018.

[142] Noha Radwan, Abhinav Valada, and Wolfram Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *arXiv preprint arXiv:1804.08366*, 2018.

[143] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *international conference on 3d vision*, pages 239–248, 2016.

[144] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian D Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016.

[145] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *neural information processing systems*, pages 1161–1168, 2005.

[146] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009.

[147] Derek Hoiem, Alexei Efros, and Martial Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 24:577–584, 2005. doi: 10.1145/1186822.1073232.

[148] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *Computer Vision and Pattern Recognition*, pages 1253–1260. IEEE, 2010.

[149] Lubor LadickÃ¡, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *computer vision and pattern recognition*, pages 89–96, 2014.

[150] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth extraction from video using non-parametric sampling. In *european conference on computer vision*, pages 775–788, 2012.

[151] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2014. ISBN 1063-6919. doi: 10.1109/CVPR.2014.97.

[152] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *neural information processing systems*, pages 2366–2374, 2014.

[153] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *neural information processing systems*, pages 1097–1105, 2012.

[154] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[155] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *international conference on computer vision*, pages 2650–2658, 2015.

[156] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[157] Bo Li, Yuchao Dai, and Mingyi He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition*, 83: 328–339, 2018. ISSN 0031-3203.

[158] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2018. ISSN 1051-8215.

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

[160] Yevhen Kuznietsov, JÃűrg StÃijckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6655, 2017.

[161] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017.

[162] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.

[163] Philipp KrÃd'henbÃijhl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[164] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.

[165] Marvin TT Teichmann and Roberto Cipolla. Convolutional crfs for semantic segmentation. *arXiv preprint arXiv:1805.04777*, 2018.

[166] Li Bo, Shen Chunhua, Dai Yuchao, A. van den Hengel, and He Mingyi. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1119–1127, 2015. ISBN 1063-6919. doi: 10.1109/CVPR.2015.7298715.

[167] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017.

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

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

[170] Yevhen Kuznietsov, JÃűrg StÃijckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2215–2223, 2017.

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

[172] Fangchang Mal and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE Int. Conf. Robot. Autom.*, pages 1–8. IEEE, 2018.

[173] Yiyi Liao, Lichao Huang, Yue Wang, Sarath Kodagoda, Yinan Yu, and Yong Liu. Parse geometry from a line: Monocular depth estimation with partial laser obser-

vation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5059–5066. IEEE, 2017.

[174] Cesar Cadena, Anthony R Dick, and Ian D Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. In *Robotics: Science and Systems*, volume 5, page 1, 2016.

[175] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 175–185, 2018.

[176] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[177] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization ? *Comput. Vis. Pattern Recognit.*, pages 1637–1646, 2017. doi: 10.1109/CVPR.2017.654.

[178] Junqiu Wang, Hongbin Zha, and Roberto Cipolla. Coarse-to-fine vision-based localization by indexing scale-invariant features. *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, 36(2):413–422, 2006. ISSN 10834419. doi: 10.1109/TSMCB.2005.859085.

[179] Ryan W Wolcott and Ryan M Eustice. Visual localization within lidar maps for automated urban driving. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183. IEEE, 2014.

[180] Alexander D Stewart and Paul Newman. Laps-localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds. In *2012 IEEE International Conference on Robotics and Automation*, pages 2625–2632. IEEE, 2012.

[181] Peer Neubert, Stefan Schubert, and Peter Protzel. Sampling-based methods for visual navigation in 3d maps by synthesizing depth images. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2492–2498. IEEE, 2017.

[182] Yuquan Xu, Vijay John, Seiichi Mita, Hossein Tehrani, Kazuhisa Ishimaru, and Sakiko Nishino. 3d point cloud map based vehicle localization using stereo camera. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 487–492. IEEE, 2017.

[183] Youngji Kim, Jinyong Jeong, and Ayoung Kim. Stereo camera localization in 3d lidar maps. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, 2018.

[184] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435, 2009.

[185] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3971–3978. IEEE, 2013.

[186] Tim Caselitz, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Monocular camera localization in 3d lidar maps. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1926–1931. IEEE, 2016.

[187] Jiali Bao, Yanlei Gu, Li-Ta Hsu, and Shunsuke Kamijo. Vehicle self-localization using 3d building map and stereo camera. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 927–932. IEEE, 2016.

[188] P. Ranganathan, J. B. Hayet, M. Devy, S. Hutchinson, and F. Lerasle. Topological navigation and qualitative localization for indoor environment using multisensory perception. *Robotics and Autonomous Systems*, 41(2):137–144, 2002.

[189] Hongtai Cheng, Heping Chen, and Yong Liu. Topological indoor localization and navigation for autonomous mobile robot. *IEEE Transactions on Automation Science and Engineering*, 12(2):729–738, 2015.

[190] D. M Bradley, R Patel, N Vandapel, and S. M Thayer. Real-time image-based topological localization in large outdoor environments. In *Ieee/rsj International Conference on Intelligent Robots and Systems*, pages 3670–3677, 2005.

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

[192] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

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

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

[195] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[196] Scott M Thede and Mary P Harper. A second-order hidden markov model for part-of-speech tagging. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 175–182. Association for Computational Linguistics, 1999.

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

[198] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, volume 1, page 4, 2012.

[199] Jürgen Wolf, Wolfram Burgard, and Hans Burkhardt. Robust vision-based localization by combining an image-retrieval system with monte carlo localization. *IEEE transactions on robotics*, 21(2):208–216, 2005.

[200] Jürgen Wolf, Wolfram Burgard, and Hans Burkhardt. Robust vision-based localization for mobile robots using an image retrieval system based on invariant features. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 359–365. IEEE, 2002.

[201] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2816–2823, 2013.

[202] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[203] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[204] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[205] Mohammad Norouzi, David J Fleet, and Ruslan R Salakhutdinov. Hamming distance metric learning. In *Advances in neural information processing systems*, pages 1061–1069, 2012.

[206] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[207] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[208] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2011.

[209] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[210] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *null*, pages 1735–1742. IEEE, 2006.

[211] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[212] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[213] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[214] Qing Li, Jiasong Zhu, Rui Cao, Ke Sun, Jonathan M Garibaldi, Qingquan Li, Bozhi Liu, and Guoping Qiu. Relative geometry-aware siamese neural network for 6dof camera relocalization. *arXiv preprint arXiv:1901.01049*, 2019.

[215] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016.

[216] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[217] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[218] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

[219] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014.

[220] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2014.

[221] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *International Conference on Neural Information Processing Systems*, pages 2366–2374, 2014.

[222] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[223] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1119–1127, 2015.

[224] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5506–5514, 2016.

[225] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015.

[226] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017.

[227] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2017.